Sapporo HyperLab

Open House 資料

'95 22 Mar.

Sapporo HyperLab

Open House TimeTable

22th March 1995

10:00–11:00	Project1 by Tanaka & Takebe
11:00–12:00	Project2 by Ted & Ohkubo
12:00–14:00	Lunch
14:00–16:00	Demo.
16:00–18:00	Free Discussion & Party

Digitized by the Internet Archive
in 2025

https://archive.org/details/openhousesapporo01tedn

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
ISEC94-16(1994-09)

IntelligentPadと超流通

田中 譲

北海道大学工学部電気工学科

札幌市北13西8

あらまし　本稿では、IntelligentPadの開発を通じて、著者のIntelligentPadに対する意味付けがどのように変化してきたかを、メディア·ツールキット·システムからシンセティック·メディア·システム、さらにはミームメディア·システムへの変遷を追って述べ、IntelligentPadがどのように機能強化されてきたか、システムと設計者の共進化の過程を振り返る。さらに、ミームメディアにとって重要な、ネットワークを介した流通と市場(いちば)システムの提供に関して、WWWとMosaicを用いたワールド·ワイドな市場システムの構築と、IntelligentPadをInternetのフロントエンドとした市場システムの構築について概説し、超流通型の新しい課金、請求、支払システムについて述べる。

和文キーワード　メディア·アーキテクチャ、マルチメディア、ソフトウェア部品、超流通、WWW、Mosaic

IntelligentPad and Super Distribution

Yuzuru TANAKA

Electrical Engineering Department, Hokkaido University

Sapporo, 060 JAPAN

Abstract　The IntelligentPad system was first developed as a media toolkit system, then it was improved to become a synthetic media system, and is now aiming to become a meme media. It has been coevolving together with its architect's understanding of the system. This paper first describes this coevolution process, and in the latter half it describes how the IntelligentPad system can provide new media called pads for the world-wide distribution of various intellectual resources through the Internet. It proposes two systems. The first one uses Mosaic as a catalog of pads to form a world-wide market place of pads. The second one uses the IntelligentPad itself as a frontend of the Internet to form a functionally enriched world-wide market place of pads. These also requires new schemes and new systems for the accounting, the billing and the payment.

英文 key words　media architecture, multimedia, software components, super distribution, WWW, Mosaic

1 はじめに

IntelligentPadの研究の発端は1987年の初頭に遡る。いろいろなドキュメントやツールを部品化して、エンドユーザがこれらを自在に組み合せて使える環境を作りたいというのがそもそもの目的であった。当初は、部品数がある程度増えれば飽和するであろうと安易に考えていた。研究が進み、色々なものが簡便に作れるようになってくると、このような考えが大きな誤りであることが明確になってきた。メディアは表現を伴うわけで、人類の文化を築いてきた表現の範囲にそもそも限界がある筈がなかったのである。では、どのようにしてエンドユーザに多様な部品を提供すればよいのだろうか。その答えがミーム·メディアへの発展であった。

パッドをあらゆる知的資源の交換流通のためのメディアととらえ、パッドをミーム·メディアへと発展させるためには、ミーム·メディアとしてのパッド·プールの構築をどのように促進するかが課題となる。IntelligentPadのカーネルシステムをマルチプラットフォーム対応にすることにより、異なるハードウェア上で稼働するIntelligentPadシステムの間でパッドを自在に交換し利用することは可能となる。

クロス·プラットフォームのパッドの流通交換機能の提供は、オン·ラインならびにオフ·ラインでのパッドの流通と再利用を可能にするが、ミーム·メディアとしてのパッド·プールの構築はこれらの機能の提供のみでは不充分である。パッド·プールの構築には、あらゆるパッドをブラウジングすることができ、パッドの売買や交換が行える場を提供する必要がある。このような場は市場やバザールに相当するものであり、これをInternetのような世界規模のネットワーク上に構築する必要がある。

本稿では、IntelligentPadの開発を通じて、著者のIntelligentPadに対する意味付けがどのように変化してきたか、それに応じてIntelligentPadがどのように機能強化されてきたか、システムと設計者の共進化の過程を振り返るとともに、ミームメディアとしてのIntelligentPadにとって重要な、ネットワークを介した流通と市場の提供について概説する。

2. IntelligentPad開発の動機

IntelligentPadシステムの開発プロジェクトは1987年初頭に始まった。当時はパーソナル·コンピュータの普及が急速に進んだ時期である。パーソナル·コンピュータの一般ユーザへの普及は、パーソナル·コンピュータの主たる利用形態が、自身でプログラムを開発して利用するという形態から、ワード·プロセッサ、スプレッド·シート、チャート·ツール、ドローイング·ツールなどの市販のツール·システムを用いて文書作成や数票処理に利用するという形態へと転換したことを物語っていた。これらのツール間での連携機能の提供も行われ、統合ビジネス·ソフトウェア·システムが市場に出るようになってきていた。しかし、いずれの統合ビジネス·ソフトウェア·システムも、利用者が新しく別のツールを付加することは不可能であり、ツール間の連携も機能と表現が予め想定されている定形業務に対応するものに限られていた。

何らかの処理をしてその結果を文書にまとめたいとする。殆どの場合、それに要する基本機能は利用可能なソフトウェアのどれかには要素機能として実現されている。しかし、利用者自身がこれらを取り出し必要とする処理と表現を実現できるように組み換えて連携させることは絶望的に困難であるか、本質的に不可能である。このことこそ、利用者が日常最も悩まされている典型的な問題である。

この問題の解決には、種々の機能をエンド·ユーザが直接取り扱え、それらの間の自在な連携をエンド·ユーザが直接定義できることが必要である。エンド·ユーザが自身で構成できることが望まれるシステムとは、決してDBMSやコンパイラ、あるいは数値計算プログラムのようなサーバシステムではない。これらのサーバシステムに対するクライアントシステムこそ、エンド·ユーザの多様なニーヅに答えるべく、エンド·ユーザ自身による臨機応変な構成を必要とする。

クライアント·システムは単なるGUIではない。エディタはGUIのみでは構成できない。エディタは、テキストや数票や図面を表示する一枚の紙の形状をしたウィンドウというディスプレイ·オブジェクトと、それに対するユーザの直接操作が発するユーザ·イベントに応じて文書編集処理や数表計算処理を行う内部プログラムとからなっている。このように一枚の紙の形状をしたディスプレイ·オブジェクトと内部プログラムとが一体化したオブジェクトは、単なるGUIオブジェクトとは区別するべきである。著者はこれをメディア·オブジェクトと名付けた。

3. メディア·ツールキットとしてのIntelligentPad

当時の著者の目論見は、百種類余りのメディア·オブジェクトを用意し、エンドユーザにこれらの機能連携の自在な定義を許すことにより、マルチメディア文書と各種のツールをエンドユーザが自在に合成できるシステムを開発することにあった。そのために必要なメディア·オブジェクトの種類は百種類余り

で十分であろうと漠然と考え、必要にして十分なコンストラクション・セットを明らかにしそれを提供することを目指していた。メディア・オブジェクト間の連携は、紙に紙を貼るように、ディスプレイ上でメディア・オブジェクトにメディア・オブジェクトを貼ることにより定義する方法を採用した(1)。

一般に、アプリケーション間の連携をGUIレベルで定義する方法には4種類がある。カット・アンド・ペーストとドラッグ・アンド・ドロップとリンキングと埋め込みの4種類である。リンキングは、アプリケーションを表すアイコンやGUI部品の間に、データやメッセージ通信の経路が結線によって定義される機能連携定義法である。埋め込みでは、アプリケーションはすべてディスプレイ上で一枚の紙の形状を持ち、紙に紙を貼ってレイアウトを行うように貼り合わせによって2枚の紙に対応するアプリケーション間の連携を定義する。共通台紙上にスプレッド・シートやチャートを貼ることによりこれらのデータを連動させることができる。

著者はマルチメディア文書と各種ツールを同様に扱うことにこだわった。両者の部品を統一的にパッドと呼ばれるメディア・オブジェクトによって表現することにした。メディア・オブジェクトの連携をマルチメディア文書の編集と同様の操作で定義可能にするために、埋め込みによる連携定義法を採用した。この頃のシステムを著者はメディア・ツールキット・システムと呼んでいる(1, 2)。

メディア・オブジェクトの連携に埋め込みによる連携定義法を採用したメディア・ツールキット・システムは、マルチメディア複合文書の自然な拡張である。1980年代のデスクトップ・パブリシング(DTP)はテキスト中に図表や図面、画像を自在に配置した複合文書の定義と直接操作を可能にした。オブジェクト指向の導入により、プログラムも複合文書の要素として埋め込むことが可能になった。このような試みは、複合文書が、ユーザ操作に反応して状態変化をおこす要素も持つように一般化されていった。このような複合文書をリアクティブ・ドキュメントという。

80年代中頃に、オブジェクト指向の合成プログラミング・パラダイムがウィンドウ・システムの種々のGUIツールキットを生んだ。このような背景を受けて、リアクティブ・メディアにツールキット化を適用したものがメディア・ツールキット・システムIntelligentPadである。

4. シンセティック・メディアとしてのIntelligentPad

メディア・ツールキット・システムとしてのIntelligentPadの開発を進める内に、メディア・オブジェクト部品の種類が百種類余りで十分であろうという考えが大きな誤りであることが明らかになってきた。1989年頃のことである。

さらに、マルチメディア文書の構成要素や各種ツールが統一的にパッドによって表現されるのみならず、DBMSや電子メイルなどの各種サーバシステム、あらゆるアプリケーション・システム、さらにはそれらを統合的に用いるための統合環境を与える場までもが、パッドとして統一的に表現され得ることに気付いた。それらの多くは、白紙のパッドとして表現される。意味のあるGUIインタフェースを持たないオブジェクトに対しても白紙のパッドによる表現を与えることにより、埋め込みによる連携定義を可能にしたのである。

このような拡張により、多様な部品パッドを連携相手を予め想定しないジェネリックなパッドとして定義しておくことが重要になり、パッド間の連携のためのメッセージ交換インタフェースの標準化が必要となった。その結果、パッドの論理的構造をスロット・リストとし、パッド間の連携をset、gimme、updateの3種類の標準メッセージを用いたスロット結合により実現する現在のアーキテクチャが確立した。著者はこのようなアーキテクチャをシンセティック・メディア・アーキテクチャと名付けた(3, 4)。

5.ミーム・メディアとしてのIntelligentPadへの発展

パッドの種類に制限がなく、多様なツールをパッドとして実現できるとすると、誰がそのような多様なパッドを提供することになるのであろうか?基本部品パッドの開発はプロの仕事であるから、企業や大学のソフトウェア開発者によって開発されることになる。開発グループは一社に特定できるであろうか?そのようなことをすれば、IntelligentPadと同じようなアーキテクチャの競合システムが複数の会社から製品化され、それぞれが相互に連携することができない互換性のない部品を提供することになるだろう。それでは結局は現在の状況と同じ問題を引き起こす。この問題を解決するには、多数の部品パッド開発会社が、相互の部品の連携を保証しつつ多様な部品パッドを提供し得るように、IntelligentPadのカーネル・システムを共通社会基盤にまで発展させる必要がある。

このような考えから、1991年頃、著者の関心は情報の生態系を考慮したシステム・アーキテクチャへと広がる。IntelligentPadを単に部品パッドの提供会社のための共通社会基盤へと発展させるだけでなく、IntelligentPadをあらゆる知的資源の「編集」と「流

[illegible]の[illegible]からさまざまな[illegible]であることが明らかになってきたこと、[illegible]のことである。

第3は、マルチメディア文書の[illegible]であ[illegible]によって表現されるものを含む[illegible]の各種データ[illegible]、ハードコピー[illegible]。さらに[illegible]のための[illegible]。[illegible]に表現されることは[illegible]。[illegible]としては、[illegible]を行うな[illegible]によってこれを表現することにより、[illegible]を可能にしたのである。

このような要請により、[illegible]を用いて記述しておくことが可能となり、パッド間の連携のためのメッセージ[illegible]インタフェースの標準化が必要となった。その結果、パッド間の結合関係をスロット[illegible]として、パッド間の[illegible]をset、gimme、updateの3種類の標準メッセージを用いたスロット結合により定義する現在のアーキテクチャが確立した。筆者はこのようなソフトウェアキットをダイナミック・ハイパーテキストと名付けている。

3. ホームメディアとしてのIntelligentPadへの要請

パッドの[illegible]に制限がなく、新規なツールをパッドとして追加できるとすると、[illegible]ようなパッドを[illegible]することにより[illegible]。[illegible]などの[illegible]は、[illegible]できる[illegible]の[illegible]であり、[illegible]が[illegible]ことができる。[illegible]IntelligentPadを[illegible]なアーキテクチャの[illegible]が[illegible]。それぞれが[illegible]することが[illegible]の[illegible]を[illegible]することになるだろう。それぞれが[illegible]を行う[illegible]ことになる。この[illegible]を[illegible]するには、[illegible]IntelligentPadのサーバーシステムを[illegible]にまで[illegible]する必要がある。

このようなことから、1991年頃、筆者は[illegible]のために[illegible]システムアーキテクチャとしてIntelligentPadを[illegible]のための[illegible]ことになる[illegible]IntelligentPadを[illegible]

文字分であろうと[illegible]。[illegible]して[illegible]システムをウィンドウセットを明らかにした[illegible]を提供することを[illegible]している。[illegible]の範囲は、紙に描を描くように、ディスプレイ上で[illegible]ことにより[illegible]する方法を提供した。

一般に、アプリケーション側の[illegible]をGUIで定義する方法には4種類がある。[illegible]ツールキット、[illegible]の4種類である。[illegible]は、アプリケーション[illegible]からGUI部品の[illegible]、データ[illegible]によって[illegible]である。[illegible]では、アプリケーションは[illegible]を行う[illegible]ことによって[illegible]。[illegible]ことによりこれらのデータを[illegible]することができる。

[illegible]方式とを[illegible]ことに[illegible]。両者の[illegible]を統一的に[illegible]によって[illegible]することにした。[illegible]の[illegible]と[illegible]可能にするために、[illegible]による[illegible]を採用した。この[illegible]システムを筆者はメディア・ツールキット・システムと呼んでいる[11]。

[illegible]の[illegible]に[illegible]によるデータ・ツールキット・システムは、[illegible]である。1980年代の[illegible]は、[illegible]を[illegible]に[illegible]した。[illegible]により、[illegible]が[illegible]ことが可能になった。このような[illegible]は、[illegible]として[illegible]が一般化されている。このような[illegible]と[illegible]。

80年代中頃に、[illegible]のシステムの[illegible]のGUIツールキットを[illegible]た。このような背景を受けて、[illegible]はツールキット化を[illegible]したものがメディア・ツールキット・システムIntelligentPadである。

4. シンセティックメディアとしてのIntelligentPad

メディア・ツールキット・システムとしてのIntelligentPadの開発を始めた[9]。[illegible]

通」と「管理」の基盤アーキテクチャとして見直す見方が次第に育って行った。部品パッドの提供会社だけでなく、パッドの利用者であり、編集者ともなりうるエンドユーザをも含めた社会の基盤アーキテクチャとして見直したのである。こうして行き着いた所が1993年より提唱しているミーム·メディアの概念である(5)。ミーム·メディアは社会において流通され編集されるあらゆる知的資源からなる文化の発展を支援するメディアである。

文化とは、外在化された知識や情報が共有され、引用され編集されて新しい知識や情報を生み、それらが順に積み上げられたものと考えることができる。知識や情報の外在化によって、個体が獲得した知識や情報が体外に記録·保存され、コミュニテイにおいて流通するようになり、共有されるようになる。そうすると、保存されている情報の引用や再利用が可能になり、それを用いて新しい知識や情報を編集することが可能になる。新しい知識や情報は同様に共有され、コミュニテイが保存する共有知識の上に積み上げられる。これが文化といわれるものである。この過程で、我々は情報の(1)外在化、(2)記録·保存、(3)流通、(4)共有、(5)引用、(6)編集の6つの情報活動を行っている。社会において流通され編集されるあらゆる知的資源からなる文化の発展を支援するには、これら6つの情報活動をすべて統合的に支援できる新しいメディアをコンピュータ上に定義し、そのメディア文化を育てることが必要である。このようなメディアは、編集と流通の機能を持つことが特徴である。

社会における文化の編集と流通の媒体としてのメディアは、遺伝子と同様に編集、複製が可能で、社会によって自然淘汰されるという特性を持ち、R.ドーキンスのいうミーム(文化遺伝子)の特性を持つミーム·メディアであるといえる。遺伝子は、組み替えと、増殖と、淘汰によって進化してきた。R.ドーキンスは同様の機構が文化の進化にも働いているのではないかと考え、ミーム(文化遺伝子、meme)という概念を提案した。組み替えは編集に、増殖は複製と流通に、淘汰は引用や市場原理による篩分けに対応する。

ミーム·メディアは、文化として積み重ねられていく社会の共有知的資産を自在に編集し複製することを可能にすることにより、知的資産の急速な増大を約束する。さらに、社会を構成する様々な人による利用と評価により、知的資産の淘汰が行われ、ミーム·プールとしての知的資産の集合の急速な進化がもたらされる。その結果生まれるのは、現在の消費文化と同様の多様な知的資源の氾濫である。知的資源の氾濫は進化には必須である。知的資源の氾濫によって充分に大きなミーム·プールが形成されるからである。問題はこの氾濫の中からある目的に適った資源をいかにして見つけ出すかである。ここに、知的資源の管理の必要性が生じる。利用者を消費者に、知的資源を商品に読み替えれば、これらの問題は現在の消費文化社会で日々考えられている問題そのものである。ミーム·メディアは知的資源の消費文化と消費文化社会を生むことになる。

知識の流通のためのメディアの開発の必要性は、1986年にXerox PARCのM. Stefikによって指摘されている(6)。彼もまたミームについて触れ、ミームのような性質を持ったメディアの必要性を論じている。

6. IntelligentPadシステム

知的資源の編集·流通·管理は外在化された知的資源に対して行われる。知的資源の外在化はメディアを用いて行われる。編集·流通·管理のすべての対象となりうる単一種類のメディアが必要である。このメディアは、知的資源の部品化が可能で、部品の機能連携が部品の組合せ編集によって定義できなくてはならない。

各部品をジェネリックに定義するには、どのような相手と機能連携するかを予め仮定する訳にはいかない。したがって、部品のどのような機能に連結するのかを編集時に簡便に指示できるようなアーキテクチャが望ましい。IntelligentPadでは、丁度AV機器がコンポネント化されたときに、各機器のインタフェースが複数個のコネクション·ジャックの装備という形を取ったように、各部品の外部インタフェースをコネクション·ジャックの役目を果たすスロットを複数個定義することでモデル化する(図1)。これをスロット·リストと呼ぶ。さらに、各部品を別の部品のジャック、つまりスロットに結合できるように、ピン·プラグが一本用意されている。ピン·プラグを一

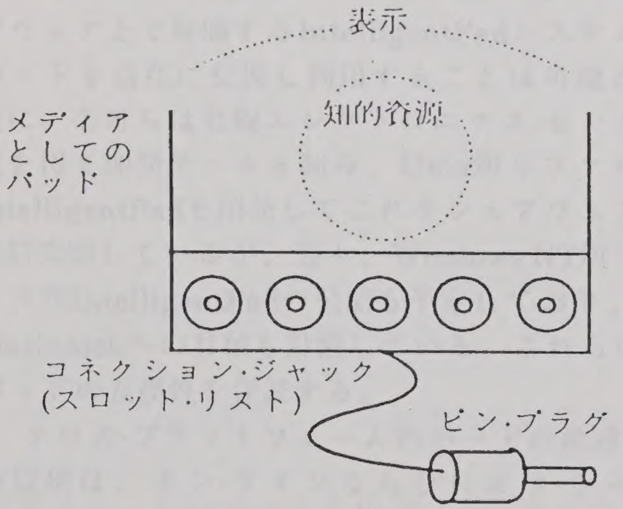
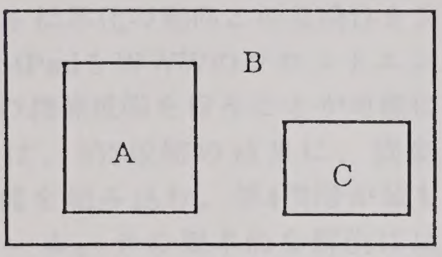


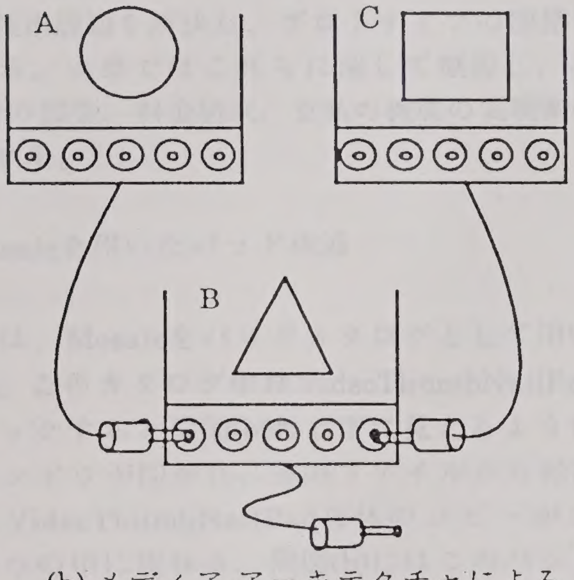
図1　パッドのジェネリックな定義を可能にする論理構造

本に制限することにより、結合構造を単純化し、各部品のモデュラリティを高め、合成パッドの再編集が容易になることをねらっている。

各スロットのアクセス·メッセージはset <slot_name> <value> とgimme <slot_name>の2種類に限定した。各スロットには、setとgimmeに対する手続が定義される。さらに、部品Aを部品Bに貼るときには、AのBに対する結合として、Bのスロットの内のただ一つを選ぶことができ、Bはこの選ばれたスロットに対してのみsetないしはgimmeメッセージ、あるいは両方を送ることができる(図2)。Aのプログラミングにおいて、結合相手のマスター·パッドに対してsetとgimmeを発する際には、スロット名をパラメータに指定しないでおく。プログラムの実行の際には、貼り合わせの際に指定されたマスター·パッドのスロットの名前がsetやgimmeメッセージのパラメータに付加されてこれらのメッセージがマスターパッドへと送出される。これにより、各パッドの動作を、貼付によって結合される相手のパッドに依存することなくジェネリックに定義することができる。



(a) パッドの貼り合わせ



(b) メディア·アーキテクチャによるパッドの連携

図2　メディア·アーキテクチャの採用によるパッドの連携

マスター·パッドのBからAには、パラメータなしのupdateメッセージを送ることができる。Bがupdateをどう用いるかは、Bのプログラマの責任に任される。Aが送られたupdateに対してどう反応するかもまた、Aのプログラマの責任に任される。

貼り合わされた2枚のパッド間に許されるメッセージには、set、gimme、updateの3種類のメッセージの外にも、move、copy、resizeなど、すべてのパッドに共通の属性を変更するメッセージなどのように、いずれのパッドにも送ることができるいくつかのメッセージが含まれる。

貼りつけるパッドをマスターパッドの適切なスロットに対応付けるには、サブパッドをマスターパッドの上に移動したのち、コマンド·メニューを開いてconnectionを選び、コネクション·シートを開いて指定する。サブパッドにはこのスロット名が保持される。

新しいパッドは、プログラムで定義するか、既にあるパッドを貼り合わせて合成することができる。パッドは自在にコピーすることができる。コピーによってのみ、新しいパッドが生成される。パッドはすべて永続性を持つ。陽に消去しない限り、パッドは存在し続ける。IntelligentPadでは、プリミティブ·パッドも合成パッドも、どのパッドに対しても常に共有コピーを即座にとることができる。

7. パッドの市場アーキテクチャ

7.1　Internetを用いたパッド流通

パッドをあらゆる知的資源の交換流通のためのメディアとととらえ、パッドをミーム·メディアへと発展させるためには、ミーム·メディアとしてのパッド·プールの構築をどのように促進するかが課題となる。IntelligentPadのカーネルシステムをマルチプラットフォーム対応にすることにより、異なるハードウェア上で稼働するIntelligentPadシステムの間でパッドを自在に交換し利用することは可能となる。既に、著者らは札幌エレクトロニクス·センターの支援を得て開発チームを組み、Unix用リファレンス版IntelligentPadを開発してこれをシェアウェアとして無償公開しているが、近々、Windows NT用リファレンス版IntelligentPadの公開も予定しており、OS/2やMacintoshへの移植も計画している。これらはすべてパッドの互換性を保証する。

クロス·プラットフォームのパッドの流通交換機能の提供は、オン·ラインならびにオフ·ラインでのパッドの流通と再利用を可能にするが、ミーム·メディアとしてのパッド·プールの構築はこれらの機能の提供のみでは不充分である。パッド·プールの構築には、あらゆるパッドをブラウジングすることができ、パッドの売買や交換が行える場を提供する必要が

ある。このような場は市場やバザールに相当するものであり、これをInternetのような世界規模のネットワーク上に構築する必要がある。

著者は、これを4段階に分けて実現しようとしている。第1段階では、WWWとMosaicを用いてパッドの流通を行う。Mosaicをパッドのカタログとして用いる。世界中のサーバに登録されているパッドを捜し出すのにこのカタログを用いる。

第2段階では、Mosaicの代わりにIntelligentPadシステムをInternetのフロントエンドとして用いる。

第3段階では、第1段階と第2段階の方法を統合する。さらに、HTMLとパッドのセイブ·フォーマットとの整合性をとる。これにより、WWWおよびMosaicに関する標準化の動向との互換性を失うことなく、IntelligentPadをWWWのフロントエンドとして用いて、各種の機能拡張を行うことが可能になる。

第4段階では、第3段階の成果に、課金、料金請求、支払の機能を組み込む。第4段階が最も困難な課題を内蔵している。その根本的な解決には、超流通アーキテクチャの採用などの検討が必要である。

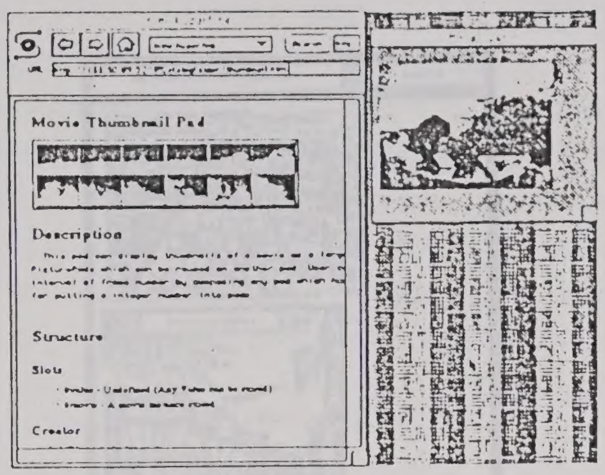
これらの各段階の内、第1段階と第2段階に関しては、既に技術課題を解決し、プロトタイプの開発を完了している。本章ではこれらに関して概説し、次章で第4段階の課金、料金請求、支払の機能の実現案の一つを紹介する。

7.2 Mosaicを用いたパッド流通

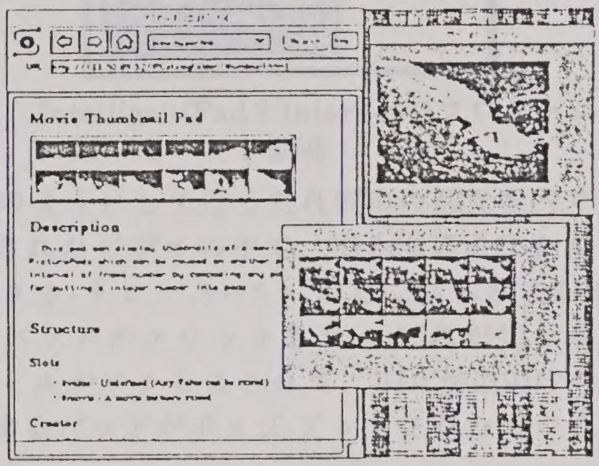
図3(a)は、Mosaicをパッドカタログとして用いた例である。このカタログ中のVideoThumbNailPadの絵をクリックすると同図(b)の右下に見えるように新しいウィンドウが開かれ、遠隔ファイルから転送されてきたVideoThumbNailPad自体のコピーがこのウィンドウの中に現れる。同図(b)にはこのパッドをユーザサイトが持つビデオに適用した結果が示されている。

Mosaicをパッドのカタログとして用いるには、対象パッドのディスプレイ·イメージをGIFファイルとして作成しておき、対象パッドの説明文書をHTMLで記述すればよい。この説明文にはGIFファイルも埋めこみ、対象パッドのディスプレイ·イメージもMosaic上に表示できるようにしておく。HTMLには他のファイルへの参照リンクを埋めこむことができるので、パッドのディスプレイ·イメージをアンカーとして、これにリンクを埋めこむことにする。このようにして得られたHTMLファイルをWWWサーバに登録しておく。

通常はリンク先ファイルとして別のHTMLファイルが参照される。このような場合には、アンカーをマウスでクリックすると、Mosaic上に他のページが表示される。Mosaicにはリンク先のファイルとして、任意のファイルを指定することができる機能が用意されており、そのファイルのファイル名の拡張子の違いに応じて、このファイルの遠隔読み出しと共にMosaicクライアント側のシステムで、このシステムに予め登録しておいた任意のプロセスを立ち上げる機能が用意されている。この機能を用いると、Mosaicをパッド·カタログとして用いることにより、ネットワークに跨るパッドの流通を行うことができる。対象パッドのセイブフォーマット·ファイルを用意し、このファイルのファイル名の拡張子をPADとする。このファイルをパッドの提供を行おうとするサイトのWWWサーバに格納する。このパッドに関する説明を含むパッド·カタログのHTMLファイルもこのサーバに格納する。このカタログのこのパッドに関する記述中には、このパッドへの参照リンクが含まれるが、このリンクをこのセイブフォーマットのファイルへのURLで実現する。そして、クライアント側のMosaicでは、ファイル拡張子がPADのとき、IntelligentPadのパッドローダを立ち上げ、遠隔アクセスされるファイルの内容を、このパッドローダに渡すことにする。



(a) Mosaicを用いたパッド·カタログの1頁



(b) 遠隔サーバより転送されたパッド

図3 Mosaicを介したパッド流通

Mosaicを用いて、あるWWWサーバからパッドカタログをアクセスし、その中のあるパッドの絵をクリックすると、この絵をアンカーとして登録してあった参照リンクがMosaicシステムに得られる。このリンクはURLで指定されており、Mosaicはこれを用いてこのパッドのセイブフォーマット·ファイルを遠隔アクセスする。このファイルは拡張子としてPADを持つので、クライアント側でIntelligentPadのパッドローダが立ち上げられ、読み出されたファイルがこのローダに渡される。ローダはこのセイブフォーマットのファイルから、対応する合成パッドを再構築する。こうしてパッドがサーバ側のサイトからクライアント側のサイトへと転送される。

7.3 IntelligentPadによるInternetのフロントエンド·システム

IntelligentPadでは、いろんな機能をもった部品パッドが用意されている。クリックされると、予め登録しておいたURLを用いて遠隔ファイルをアクセスし、そのファイルにセイブフォーマットで格納されているパッドをディスプレイ上にポップ·アップするパッドも既に用意されている。このパッドをURLアンカーパッドと呼ぶ。URLアンカーパッドはボタンパッドと同様に用いることができ、ラベルを上に貼ったり、自身を透明にすることができる。

IntelligentPadにはビデオパッドも用意されており、ビデオパッドの上でフレーム毎に位置とサイズを変化することができるオブジェクト·トレース·パッドもすでに用意されている。

Internet上のどこかに格納されている任意のパッドに対し、その格納ファイルを指し示すURLアンカーパッドが手元にありさえすれば、このURLアンカーパッドをクリックすることによって遠隔ファイルをアクセスし、手元のディスプレイ上に対象パッドをポップ·アップすることができる。得られたパッドの上に、やはりInternet上のどこかに格納されているパッドを指すURLアンカーパッドが貼られていてもよい。URLアンカーパッドは透明にできるので、文書の一部や、図表の一部をこれで覆ってアンカーの役目を果たすように指定することができる。

ビデオパッドを用い、ビデオ中の幾つかの対象物に対し、その動きと大きさの変化に応じて常にこれらを覆うように指示したオブジェクト·トレース·パッドを必要な数だけ用意し、これらのオブジェクト·トレース·パッドの上にURLアンカーパッドを貼り、各々が覆っている対象物に関する詳細情報やツールを提供するパッドの格納ファイルへのポインタの役目を果たすURLを忍ばせることができる。図4はこのようにして作った札幌市の観光案内である。空撮されたビデオの中で、旧北海道道庁を覆うオブジェクト·トレース·パッド上に貼られたURLアンカーパッドがクリックされ、これが指し示す遠隔ファイルがアクセスされて、旧北海道道庁の写真を表示するパッドがポップ·アップされたところを示している。

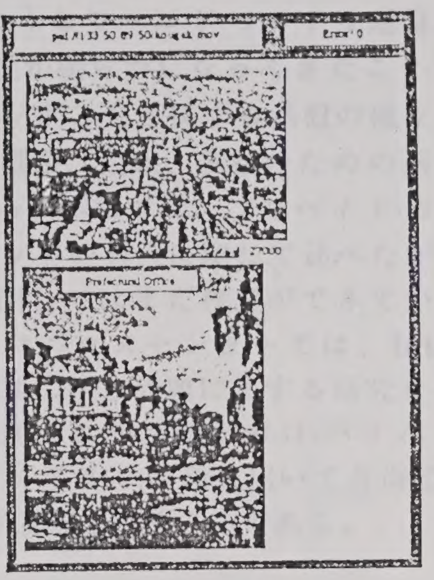


図4　IntelligentPadをInternetのフロントエンドに用いたパッド流通

このような仕組みにより、Internet上で遠隔地のサーバに格納されているビデオパッドや各種ツールパッドを次々と辿って気に入ったパッドや必要なパッドを手元に持ってくることが可能になる。このようなシステムの直接的応用としては、ネットワーク·ミュージアムやネットワーク·マルチメディア·ライブラリ、バーチャル·モールなどが考えられる。

8. パッド使用に対する課金、請求、支払

IntelligentPadの各種パッドをミームメディアとして発展させるには、各種パッドがユーザによって自由にコピーされ、再流通されなくてはならない。したがって、パッドのコピーの入手に対する課金ではなく、パッドの使用に対する課金の方法を確立する必要がある。もちろん、売り切り方式で販売されるパッドもあってよい。支払に関しては、パッドの使用前、使用時、使用後のいずれに行う方式も考えられるが、現在検討中の方式は、パッドの使用前に支払を済まし、使用時に使用料を清算するプリペイド·カード方式である。

プリペイド·カードもまたパッドとして実現する。あるベンダー製のパッドはこのベンダーの発行するプリペイド·パッドの上でのみ使用することがで

きる。対応するプリペイド·パッドがない場合にはパッドは動作しない。異なるベンダーが発行する複数のプリペイド·パッドの情報を一つのプリペイド·パッドにまとめて保持することもできる。プリペイド·パッドには総価値の上限と使用期限が記憶されている。プリペイド·パッド上でそのベンダー製のパッドを使用すると、そのパッドに予め設定されている課金方式に従った使用料がプリペイド·パッドの総価値より引き落とされる。異なるベンダー製の何枚かのパッドを用いる場合には、これらのベンダーのプリペイド·パッドを重ねて用いるか、それらを一つのパッドにまとめたプリペイド·パッドを用いる。パッド自体のコピーや他者への供与は自由であるが、プリペイド·パッドはプロテクトをかけ、ユーザがこれをコピーしたり、その内容を変更することができないようにする。このプロテクト法は定期的に変更する。

パッドに予め設定される課金方式をどう設定するかによって、売り切り、一定期間の無償使用供与、一部の機能をはずした無償使用供与、無償使用供与などにも対応でき、プリペイド·パッドをどう設定するかによって、サイトライセンス、ボリューム·ディスカウント、アカデミック·ディスカウントなどにも対応できる。スポンサー付きのプリペイド·パッドの発行も可能である。パッドそのものは無償配布も可能となり、CD-ROMを用いて出版流通経路にのせたり、放送電波を用いた配布、BBSやFTPを用いた配布などが考えられる。

あるベンダーが数社のパッドを組み合せて開発したツールを製品化する場合もある。この場合、このツールを分解不可能にすることにより、このツールの使用に応じて、このベンダーと部品ベンダーの各々のプリペイド·パッドから一定の割合で使用料を引き落とすことも可能である。

プリペイド·パッドの購入に関しては、電話回線を用いて行う方法などが考えられる。

9. 結論

IntelligentPadはこの7年半の間、その信奉者とともに共進化してきた。そして現在、関係者の多くが今後の進化の方向として、ミームメディアへの発展を共に意識するようになった。その中で、Internetを介したワールド·ワイドなパッド流通機構の構築は、Mosaicの急速な普及にも刺激され、関係者の関心を集めている。Mosaicがマルチメディア文書の流通を主として、簡単なトランザクションの送受にも拡張されようとしているのに対し、IntelligentPadをInternetのフロントエンドとして用いれば、あらゆるドキュメントとツールの流通、あらゆるトランザクションの送受を容易に実現することができる。

このようなワールド·ワイドな基盤システムの実現の可能性が明らかになってきたことにより、その上でのビジネスを支援する基盤の確立が重要となってきた。課金、請求、支払のための基盤システムの確立である。本稿では、プリペイド·カードの役割を果たすパッドの導入に関して述べたが、その保全性の確立に関してはまだ検討ができていない。札幌市エレクトロニクスセンターでは、IntelligentPadのパッド流通に伴う課題に関する研究を促進するために、本年10月より、サッポロ·ハイパーラブを設立し、テッド·ネルソン氏を招いて北海道大学と集中的に共同研究開発を行う予定である。

参考文献

(1) Tanaka, Y. and Imataki, T., IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations, *IFIP '89*, San Francisco, pp.541-546, Aug. 1989.

(2) Tanaka, Y., A Toolkit System for the Synthesis and the Management of Active Media Objects, *Proc. of DOOD*, Kyoto, pp. 76-94, Dec., 1989.

(3) Tanaka, Y., A Synthetic Dynamic-Media System, *Proc. of Multimedia Information Systems*, Singapore, pp.299-310, 1991.

(4) Tanaka, Y., A. Nagasaki, M. Akaishi, and T. Noguchi, Synthetic Media Architecture for an Object-Oriented Open Platform, *IFIP 92*, Vol III, (ed. F. H. Vogt), North Holland, pp104-110, 1992.

(5) Tanaka, Y., From Augmentation Media to Meme Media, *World Conference on Educational Multimedia and Hypermedia*, Vancouver, pp.58-63, June 1994.

(6) Stefik, M., The Next Knowledge Medium, *The AI Magazine*, 7 (1), pp.34-46, 1986.

Sapporo HyperLab

OPEN HOUSE

at 22nd March, 1995

Project 1(Short term)

director

Prof. Yuzuru Tanaka(Hokkaido Univ.)

researcher

Kiyoki Ookubo (Vision)

Kazuhide Nakajima (Fuji Xerox)

Takashi Nakajima (NTT Hokkaido)

Minoru Hidaka (Hitachi Central Laboratory)

Ryoko Kambayashi (Tecnova)

Tatsuya Kainuma (Fujitsu Aichi Engineering)

&

Yoshihiko Takebe (Fujitsu)

Written by Yoshihiko Takebe

Table of Contents

Mosaic を利用した Pad の流通 3

Hyper Text Markup Language の記述方法
Mosaic の環境設定
IntelligentPad を起動するには？
開発
  iploader
  pstogif
  save ＋ print Pad

IntelligentPad を利用した Pad の流通 7

HTML Document View Pad 8

今後へ向けて 9

Jpeg & Mpeg Pad/Modifier
Pad の課金
  超流通
  ダイヤルQ^2
  Digital Money の調査

MOSAIC を利用したPAD の流通

- NCSA-Mosaic の URL(Uniform Resource Location)機能を利用してWWW のサーバから Pad を取り出す．取り出された Pad は，Socket の機能を使い，IntelligentPad 上に表示する．

図 1 に Mosaic から Pad を IP 上にロードしたときのイメージ図を示す．

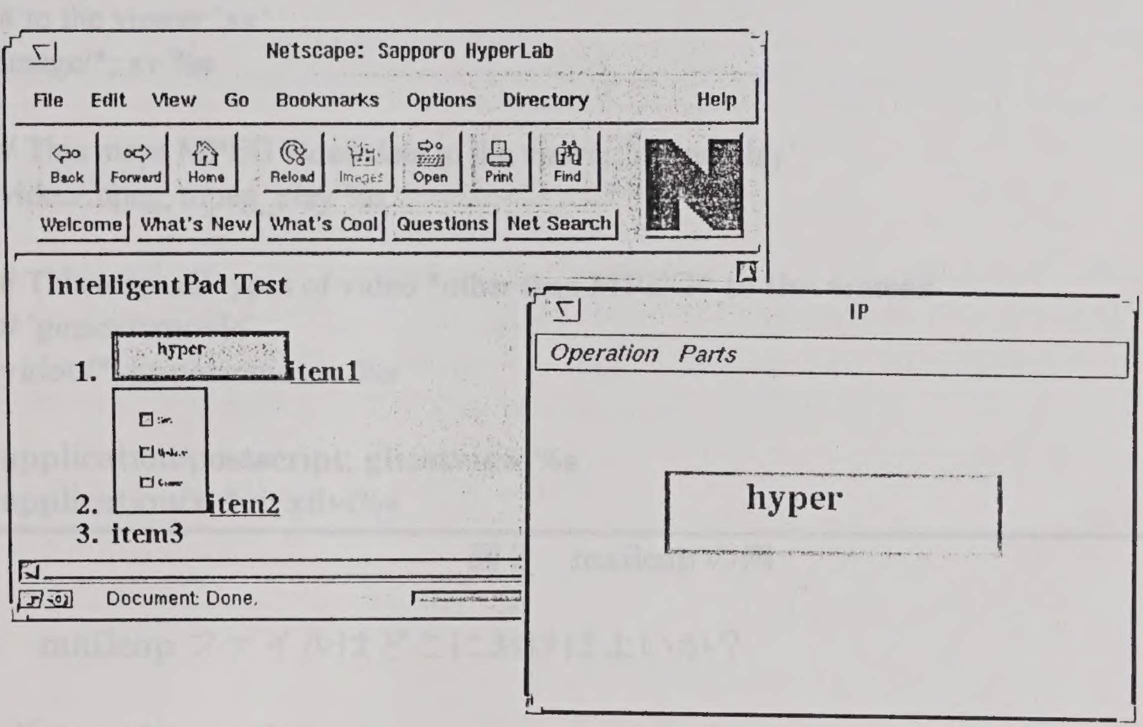

Netscape: Sapporo HyperLab
File
Edit
View
Go
Bookmarks
Options
Directory
Help
Back
Forward
Home
Reload
Images
Open
Print
Find
Welcome
What's New
What's Cool
Questions
Net Search
IntelligentPad Test
hyper
1.
item1
2.
item2
3. item3
Document: Done.
IP
Operation
Parts
hyper

図 1　Mosaic から Pad をロードした場合のイメージ図

- HyperText Markup Language (HTML)の記述方法

＜例＞

```
<A HREF="http://hyper.sec.or.jp/savePads/hyper.pad>
<IMG SRC=hyper.gif>hyper.pad</A>
```

Mosaic の環境設定

1. MIME タイプと mailcap

基本的に mailcap ファイルは，外部のビューアーのために MIME タイプのマップを作るコンフィギュレション・ファイルである．(MIME は RFC 1521.によって定義されている）例えば，マッピングしている MIME タイプ"image/gif"はイメージビューアー"xv"のものである．

図２に簡単な mailcap の例を示す.

```
# This is a simple example mailcap file.
# Lines starting with '#' are comments.

# This map all types of audio data (audio/basic,audio/x-aiff,
# etc.) to the viewer ‘showaudio’. Note that ‘%s’ means ‘put the
# datafile name here when the viewer is excuted‘.
audio/*; xv %s

# This maps all types of images (image/gif, image/jpeg,etc.)
# to the viewer ‘xv’.
image/*; xv %s

# This maps MPEG video data to the viewer ‘mpeg_play’.
video/mpeg; mpeg_play %s

# This maps all types of video *other than MPEG* to the viewer
# ‘genericmovie’.
video/*; genericmovie %s

application/postscript; ghostview %s
application/x-dvi; xdvi%s
```

図２　mailcap の例

2. mailcap ファイルはどこにおけばよいか？

グローバル mailcap:

ファイル名のデフォルトは，“/usr/local/lib/mosaic/mailcap”である．これは一般にシステム，またはサイトワイド　ビューアー　コンフィギュレーションの場所である．

パーソナル mailcap:

ファイル名のデフォルトは，環境変数 HOME の値が影響する．このデフォルトは“.mailcap”である．これは一般的にパーソナルユーザーコンフィグレーションの場所である．

3. MIME タイプ ＋ 噂？

Mosaic は，初期値によってビューアーに幾つかの MIME タイプマップを割り当てることができる．例えば JPEG には，サフィックスが jpeg,jpg,jpe でも同じ扱いをするようにこの３つが登録されている．

非公式ではあるが，実験的な MIME タイプに“x-”で開始するようになっているようである．これは，AIFF オーディオがまだ MIME タイプを登録していなかった頃の古い MIME タイプが“audio/x-aiff”である．（その後，aiff が IANA に登録して，割り当てられたのが“audio/aiff”である）

- **IntelligentPad を起動するには？**

mailcap は mime types に従ったエクスターナル　ビューアーを起動するものであり，ＩＰを起動するようにするには以下の記述を追加する．

.mimetypes ファイル

```
application/x-iploader          pad
```

.mailcap　ファイル

```
application/x-iploader; /home3/ipc/EvaSys941220/bin/iploader1 %s
```

- **Iploader**

Iploader は，IP を起動し，IP 上に Pad をロードする iploaer と，Mosaic から Pad に対し URL 発した場合に既に IP が起動されているかを監視する iploader1 からなる．Iploader は，既に Iploader から IP が起動されていれば新たに IP を起動することなしに Pad を IP 上に展開する．

図３に iploader と IP の関係を示す．

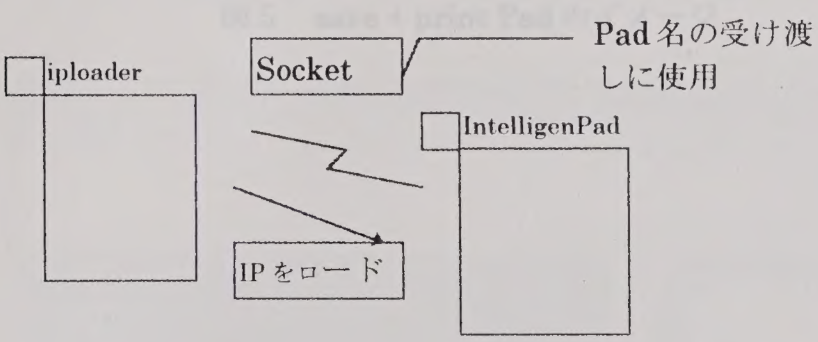

iploader
Socket
Pad名の受け渡しに使用
IntelligenPad
IPをロード

図３　iploader と IP の関係

- **pstogif**

IP の print 機能で生成された PostScript ファイルを gif 形式に変換する．図４にイメージの作成手順を示す．

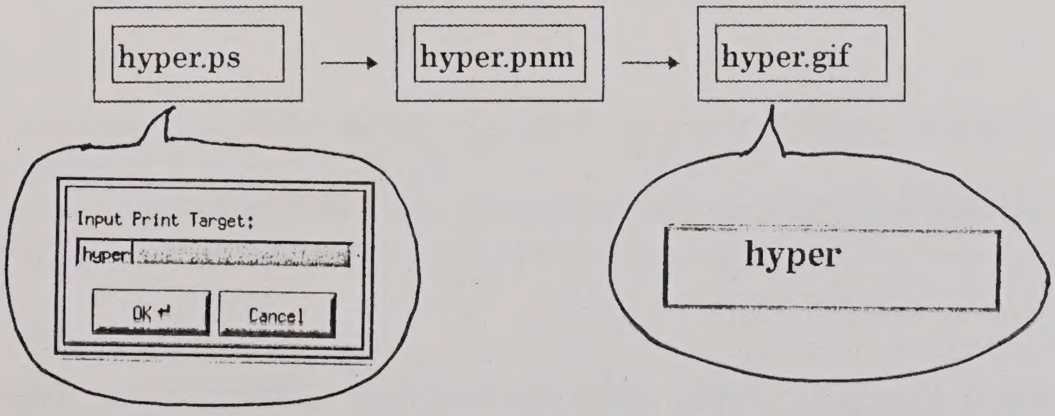

hyper.ps
hyper.pnm
hyper.gif
Input Print Target:
hyper
OK
Cancel
hyper

図４　イメージの作成手順

☛ 詳細については，World Wide Web の Helper Application インターフェース(大久保　清貴氏　ビジョン）を参照.

● Save + Prind Pad

IP では，save コマンドと print コマンドが独立しており，save と Pad をセーブすると同時に行うためには，操作が２回必要である．この操作を１回で行う Pad を作成した．この Pad は，セーブ形式と，イメージファイルを gif 形式で作成し，それぞれ“.pad “，“.gif”のサフィックスが付与され，ディスクに格納する．

図５に save + print Pad のイメージを示す．

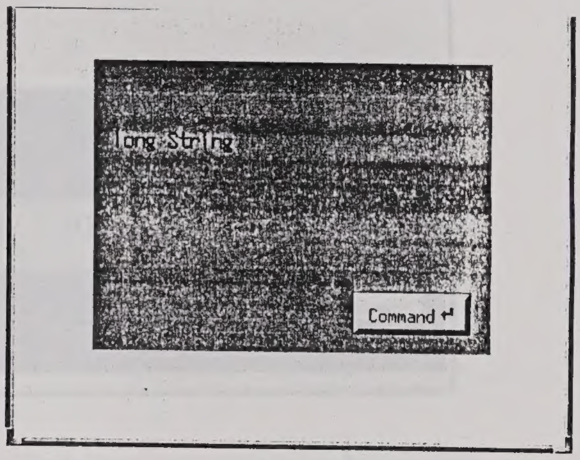


図５　save + print Pad のイメージ

IntelligentPadを利用したPadの流通

- Mosaicを利用したPadの流通方式について記述したが，IntelligentPadを利用し，WWWサーバからPadを取り出すことのできるURL Padを作成した．

図6にURL Padを示す．

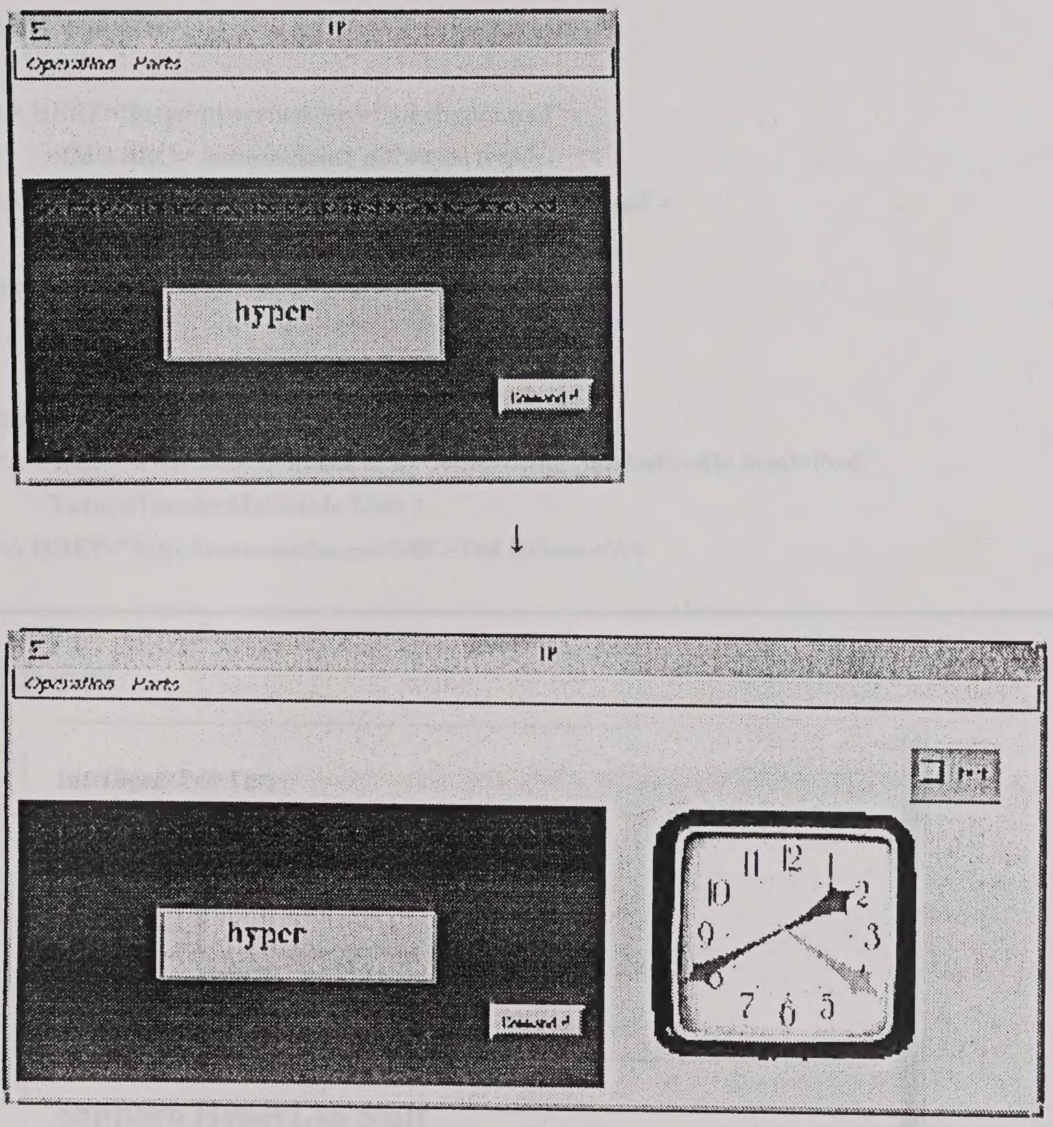

IP
Operation Parts
hyper
↓
IP
Operation Parts
hyper

図6　URL Pad

図5の例では，URLPadのボタンを押すことでWWWサーバーにあるPad(この場合は時計をクライアントのIPに取り出すことができる．
また，URL Padに“www http://mindmachime3/savePads/clock.pad”の指定をstringで行うことでURLを発するようになっている．

☛ 詳細については，URLパッドの仕組み(大久保　清貴氏　ビジョン）を参照．

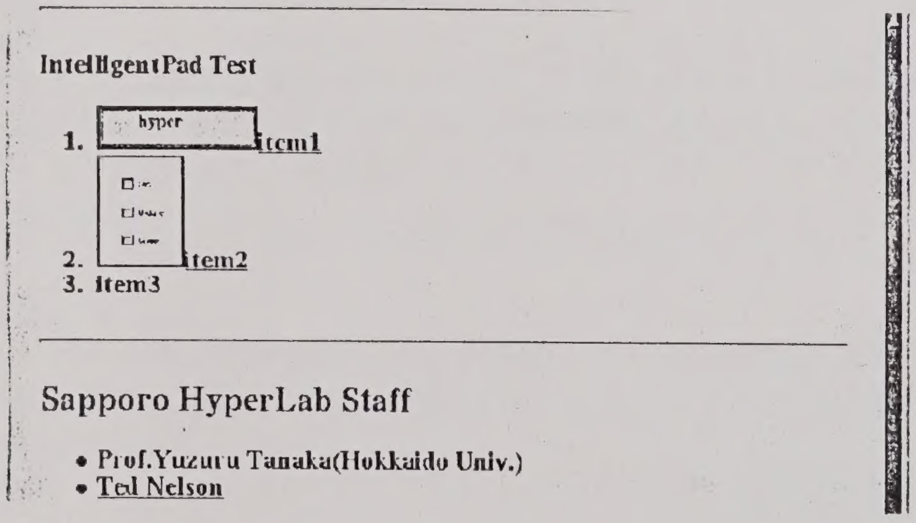
HTML Document View Pad

- IntelligentPad から HTML ドキュメントを参照するビューを Pad として用意する．

HTML 記述例

```
<H3>IntelligentPad Test</H3>
<OL>
        <LI><A HREF="http://hyper/usr/savePads/hyper.pad">
                <IMG SRC="images/item1.gif">item1</A>
        <LI><A  HREF="http://hyper/usr/savePads//.hconn.menu.pad">
                <IMG SRC="images/item2.gif">utem2</A>
        <LI>item3
</OL>
<HR>
<H2>SapporoHyperLab Staff</H2>
        <LI><A HREF="www.huee.hokudai.ac.jp/Consortium/TanakaProfile.html>Prof.
                YuzuruTanaka(Hokkaido Univ.)
        <LI><A HREF="http://sensemedia.net/849">Ted Nelson</A>
```

Mosaic での表示例



IntelligentPad Test

1. item1
2. item2
3. item3

Sapporo HyperLab Staff

- Prof.Yuzuru Tanaka(Hokkaido Univ.)
- Ted Nelson

- gif2pad.sh

Mosic で表示できる gif 形式のイメージデータを IntelligentPad が扱える tiff 形式のイメージデータに変換する．その後，作成した tiff データを Pad 形式に変換し，IP にロードする．

- URL Pad

www,gopher,archie に対してアクセスできるように URL Pad の修正（/確認）を行う．

- 課題
  1. テキストのレイアウト
  2. ダイナミックな拡大，縮小

これは，**transmodifier** を追加することで対応できそう．拡大，縮小方向のみをサポートし，回転については対応しないほうがよい（回転がはいると操作が繁雑になり，誤って回転させた場合に元に戻すのが難しい）

☛ 詳細については，html Viewer Pad(中島　和英氏　富士ゼロックス) を参照．

今後へ向けて

● JPEG & MPEG Pad/Modifier

- Interviewsのクラスライブラリに，マルチメディア操作用の拡張クラスを作成するXavier 2.0を使用してPad，またはModifierを作成する．現段階ではModifierを作成したほうが利用価値が高いのではと考えている．

- **Xavier？**

Xavierでのマルチメディアオブジェクトモデルは，以下の2つがある．

構成モデル：
メディアオブジェクトそのもののデータを扱ったり，それらを組み合わせて使用する．

再生モデル：
構成モデルに基づいて作成されたマルチメディアオブジェクトを再生する．

再生モデル内には，Viewerというオブジェクトがあり，このViewerは，Media,Contextというオブジェクトを保持している．これらの3つのオブジェクトの役割は以下のとおりである．

Media：
構成モデルにおける複合マルチメディアオブジェクトに相当するものであり，基本メディアオブジェクトそのものを管理するものと，それらを組み合わせて新たなオブジェクトを構築するためのコンポジットオブジェクトを管理するものがある．

Context：
複合マルチメディアオブジェクトを再生しているときの状況を管理する．また，実際に画面に描画したり，特定のマルチメディアデバイスを駆動するのもContextが行う．

Viewer：
マルチメディアオブジェクトを画面やスピーカーに結び付けるためのオブジェクトであり，画面上での位置情報などを持つ．また，ViewerはGlyphのサブクラスとして実現されている．

● Padの課金

ネットワーク上でのPadの課金について調査を行い，IPの課金方法を実現する.

- 超流通について

  ☛ 超流通について（上林　亮子氏　テクノバ）参照

- ダイヤルQ ²について

  ☛ ダイヤルQ ²を利用したソフトウェア流通の可能性について（中島孝志氏　NTT）参照

- Digital Moneyの調査

ecashの流れとセキュリティー

digiCash社は，金融機関にソフトウェア，ノウハウについてライセンスを供与する.

▽

個人ユーザは，DigiCashからエンドユーザー用ecashソフトウェアを無料，または安くインターネット上で入手する.

▽

ecashライセンスを持っている銀行に口座を開設する.

▽

預金でecashにして貰いたい金額を銀行に申請する.

☑ セキュリティー

ユーザのコンピューターは乱数を発生させ，ノートを作る．このノートをユーザーのコンピュータが隠蔽化する．隠蔽化処理をされた券面（ブラインドノート）は，実際の通貨の出金情報とともに銀行に送られる.

▽

銀行は依頼者の正当性をチェックし，申請金額をecashにする．この通知をインターネット上で行い，同時に口座から申請金額を引落とす.

☑ セキュリティー

銀行は，ユーザが指定した実際の通貨の申請額をユーザの口座から引落とすと同時にブラインドノートに電子署名し，ユーザーに返送する．これが銀行からオーソライズされた券面となる．この電子署名を変更できるのは，銀行のプライベートキーのみか可能であり，銀行がオーソライズしたことは銀行のパブリックキーでチェックできる.

▽

ユーザはオーソライズされた ecash を任意に分割し，インターネット上のプロバイダーから情報を入手する．

☑ セキュリティー

ユーザは返送されたブラインドノートの隠蔽を取り除き（アンブラインド），支払いを可能にする．隠蔽を取り除かれた券面は，任意に分割して支払うことができる．これは，銀行がオーソライズした金額を小額に分割することで，支払い行為のプライバシーが向上することになる．

支払いにあたっては，ユーザー側の電子署名を付ける．支払い券面の変更は，ユーザー側のプライベートキーによってのみ可能である．また，途中で改竄されておらず，支払い者の正当性チェックはユーザのパブリックキーの照合によってチェックされる．分割された券面には銀行の電子署名も付属する．

▽

プロバイダーは，ecash ライセンスを持つ銀行に口座を持っている．また ecash による支払いを受けるために DigiCash からソフトウェアを入手している．個人ユーザは，プロバイダーの持つオンラインショップから情報を入手することができる．

▽

プロバイダーが個人ユーザーから ecash を受け取る場合は，支払い者の正当性をチェックし，額面の改竄がないかを確認する．問題がなければ商取引が成立する．

☑ セキュリティー

プロバイダー等の受領者は，支払い者のパブリックキーと照合して，支払い者の正当性をチェックする．また，銀行のパブリックキーと照合して銀行のオーソライズをチェックする．

▽

プロバイダーは ecash をインターネット経由で銀行に送る．銀行は再度 ecash の正当性をチェックする．問題がなければ，プロバイダーの口座に所定のレートで換算した金額を入金する．

☑ セキュリティー

銀行ではプライベートキーによる最終的なチェックを行い，プロバイダーの口座に相当額を入金する．

▽

ecash ライセンスを持つ各国の銀行は，ecash の取り扱いについて一括で決済する．

HTML Viewer Pad

札幌HyperLab駐在
Fuji Xerox　中島和英

1　概要

Padを流通させるためには、WWWから、起動できるような仕組みを持ちたい。そのために、Padを読み込むためのアプリケーションiploaderを作りました。これは、MOSAICから、アンカータグのある文字列をクリックすると、指定したPadが読み込まれてIPが起動するようにしたシステムです。

このiploaderと、HTML（Hyper Text Mark-up Language)の文書をコンバージョンしてPadを生成するシステムと組み合わせることによって、IPでHTML Viewerの機能を持たせました。

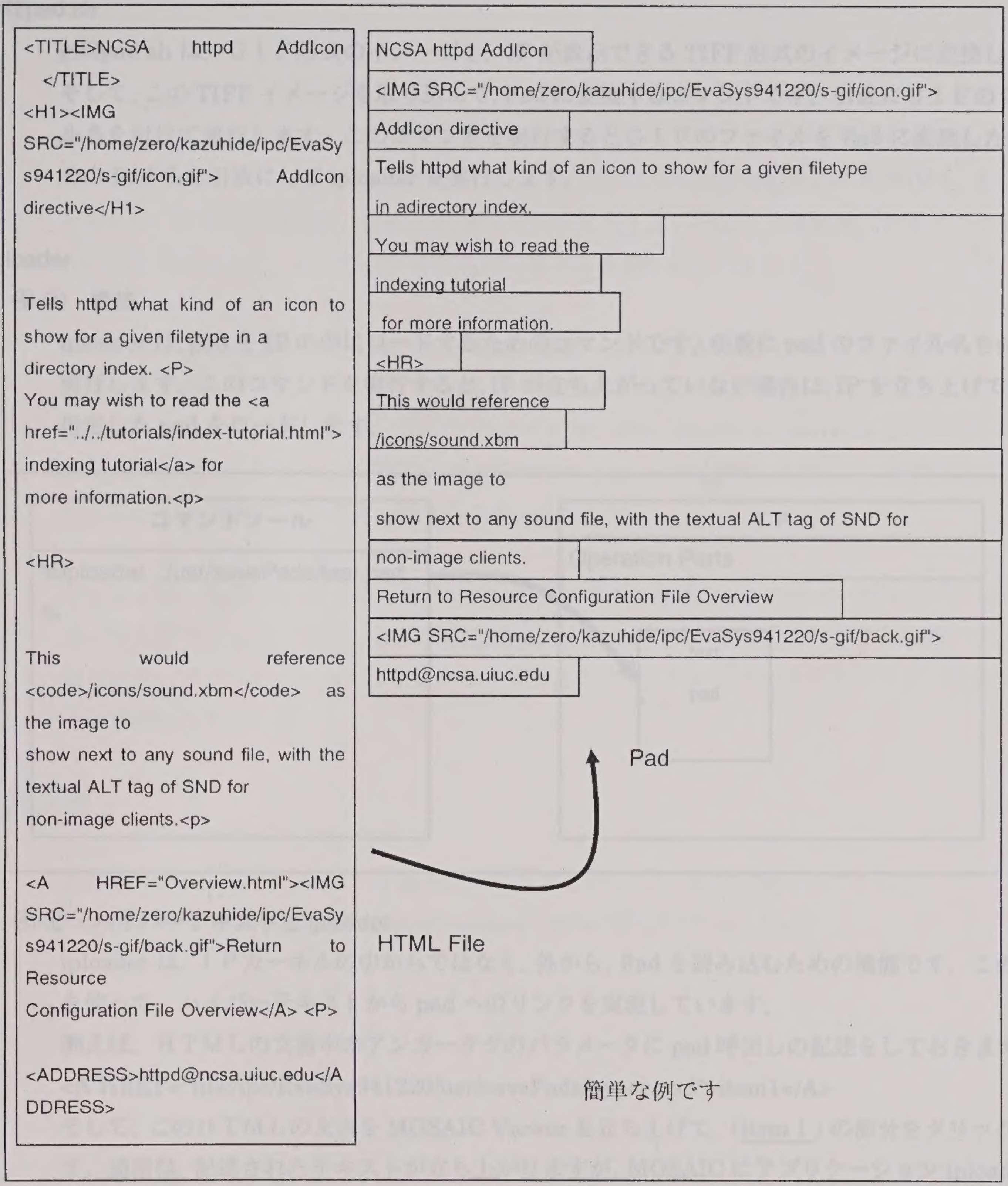
2　技術

HTML Viewerの機能は、以下の技術に分けられます。

● scan	HTMLファイルを構造解析して、テキスト、イメージに変換する
● text to pad (text2pad.sh)	テキストファイルをPadに変換する
● gif to pad (gif2pad.sh)	GIF形式のイメージファイルをPadに変換する
● iploader	指定したPadをIPに読み込む

①　scan

HTMLのファイルを構造解析して、Padに変換するコマンドです。MOSAICのビュアーの表示と同じように、各内容、見出しに分けたPadを作成して順次レイアウトしながら表示していきます。基本的には、構造解析したテキストのタグの種類によってPadを生成します。scanで分けられる種類はテキストとGIFです。この２種類をtext2pad.sh、gif2pad.shを使ってＩＰに読み込みます。以下に例として、HTMLファイルとPadの対比図を記述します。



```
<TITLE>NCSA httpd AddIcon
  </TITLE>
<H1><IMG
SRC="/home/zero/kazuhide/ipc/EvaSy
s941220/s-gif/icon.gif"> AddIcon
directive</H1>


Tells httpd what kind of an icon to
show for a given filetype in a
directory index. <P>
You may wish to read the <a
href="../../tutorials/index-tutorial.html">
indexing tutorial</a> for
more information.<p>

<HR>


This would reference
<code>/icons/sound.xbm</code> as
the image to
show next to any sound file, with the
textual ALT tag of SND for
non-image clients.<p>

<A HREF="Overview.html"><IMG
SRC="/home/zero/kazuhide/ipc/EvaSy
s941220/s-gif/back.gif">Return to
Resource
Configuration File Overview</A> <P>

<ADDRESS>httpd@ncsa.uiuc.edu</A
DDRESS>
```

NCSA httpd AddIcon
<IMG SRC="/home/zero/kazuhide/ipc/EvaSys941220/s-gif/icon.gif">
AddIcon directive
Tells httpd what kind of an icon to show for a given filetype
in adirectory index.
You may wish to read the
indexing tutorial
for more information.
<HR>
This would reference
/icons/sound.xbm
as the image to
show next to any sound file, with the textual ALT tag of SND for
non-image clients.
Return to Resource Configuration File Overview
<IMG SRC="/home/zero/kazuhide/ipc/EvaSys941220/s-gif/back.gif">
httpd@ncsa.uiuc.edu

② text2pad.sh

text2pad.sh は、テキストを Pad に変換するコマンドです。引数にテキストのファイル名を付けて実行します。このコマンドを実行するとテキストを Pad に変換したあと、その Pad 名を引数にして iploader を実行します。

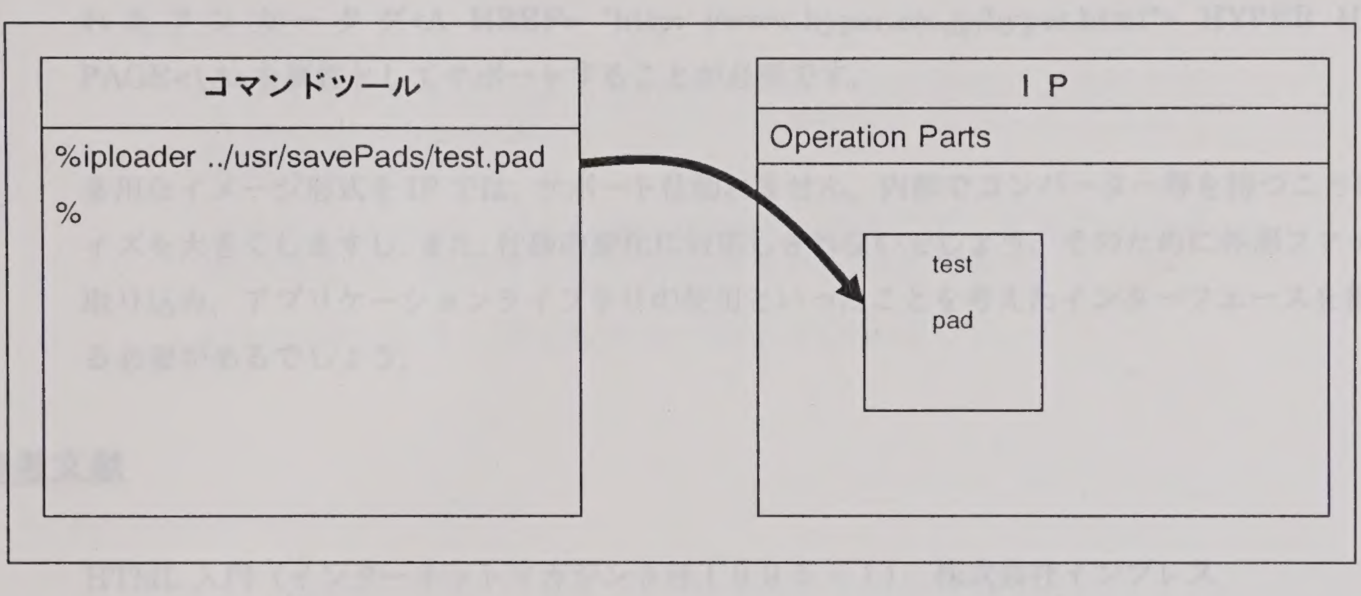
③ gif2pad.sh

gif2pad.sh は、ＧＩＦ形式のイメージを、IP が表示できる TIFF 形式のイメージに変換します。そして、この TIFF イメージを取り込んで、Pad に変換するコマンドです。引数にＧＩＦのファイル名を付けて実行します。このコマンドを実行するとＧＩＦのファイルを Pad に変換したあと、その Pad 名を引数にして iploader を実行します。

④ iploader

④-Ⓐ 機能

iploader は、pad を IP の中にロードするためのコマンドです。引数に pad のファイル名を付けて実行します。このコマンドを実行すると、IP が立ち上がっていない場合は、IP を立ち上げてから、指定した pad をロードします。



```
%iploader ../usr/savePads/test.pad
%
```

④-Ⓑ ハイパーテキストと iploader

iploader は、ＩＰカーネルの中からではなく、外から、Pad を読み込むための機能です。この機能を使って、ハイパーテキストから pad へのリンクを実現しています。

例えば、ＨＴＭＬの文書中のアンカータグのパラメータに pad 呼出しの記述をしておきます。

<A HREF="file:/ipc/EvaSys941220/usr/savePads/hyper.pad">item1</A>

そして、このＨＴＭＬの文書を MOSAIC Viewer を立ち上げて、「item 1」の部分をクリックします。通常は、記述されたテキストが立ち上がりますが、MOSAIC にアプリケーション iploader を指定することによって、リンクした pad を表示するために IP が立ち上がります。

3　課題

ＨＴＭＬは、構造化された文章の形をとっています。そのため、pad の貼り合わせによる構造で表現することができます。ただし、文章中に字体を変えたり、URL を埋め込んでおくには、その構造（属性）を解析する Pad が必要です。もし、HTML ファイルを解析するのであれば、この２点を考えて、Pad にするのか、Text 内の属性なのかを考えなければなりません。つまり、ＨＴＭＬのもつタグによって、コンバージョンを切り替える必要性があります。

HTML 形式の属性をもったテキストを表示、編集できる TextPad が必要です。現在の IP では、書体や、文字色をサポートしていません。そのために文書の表現が限られています。例えば、<B>ＴＥＸＴ<\B> で文字をボールドで現すタグや、URL (Uniform Resource Locator)が指定されたアンカータグ<A HREF= "http: //www.hyper.sec.jp/hyper.html"> HYPER HOME PAGE<\A>を属性としてサポートすることが必要です。

多用なイメージ形式を IP では、サポート仕切れません。内部でコンバーター等を持つことは、サイズを大きくしますし、また、仕様の変化に対応しきれないでしょう。そのために外部ファイルの取り込み、アプリケーションライブラリの使用といったことを考えたインターフェースを提供する必要があるでしょう。

4　参考文献

HTML 入門（インターネットマガジン３号１９９５・１）　株式会社インプレス
http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

超流通について
SUPERDISTRIBUTION

上林亮子
（株）テクノバ

１．超流通とは

１９８３年、筑波大学名誉教授の森亮一氏の発明によるもので、コンピュータを利用した、デイジタル情報・ソフトウエアの流通システムです。

超流通によれば、デイジタル情報は、自己によって所有されるのではなく、利用する度に利用条項が記録されます。

この記録に基き、適切な料金が徴収されるのです。

２．超流通の満たすべき性質

1　低コストで、大量かつ安全に供給されること

2　情報提供者の利益が保証されること

3　情報利用者の利便を阻害しないこと

３．超流通の定義

1　情報利用者は、デイジタル情報をほとんどまたは全く無料（媒体・手数料程度）で、入手することができ、情報提供者（メーカ、個人を含む）が指定した条件の下でいつでも利用できる。

2　情報提供者はその情報の利用を許可する条件（通常は料金の支払）を指定できる。

3　情報をどの程度改変できるかはシステムの設計者によって選択される。

4　以上の点について面倒な手続きを必要としない。

４．超流通の歴史

ーーーー物質文明から情報文明へーーー

これまでは、流通というと、ほとんど物流を指していた。形あるものの流通は、持てる者同士による物々交換に始まり、市場を形成する過程において、物の抽象化、即ち、貨幣が誕生するに至った。

こうして物々交換経済から貨幣経済への移行がなされ、それを支える流通システムが発展してきた。

デイジタル情報は、この物流貨幣経済の下に新たな産業製品として流通するようになった。

ところが、デイジタル情報製品と、有体物品とでは異なった性質をもつため、流通の取り決めもこれまでと同様では、不都合な点が多く見られる。

そこで、デイジタル情報製品に見合った新たな流通システムである超流通が発明された。

さらに、デイジタル情報社会に対応した法律が整備されることにより情報文明社会が健全に運営されることになる。

５．流通製品としての性質の違い

ーーー有体物ーーー

1　Ａ者からＢ者に引き渡すことにより、自己の持っていた物はなくなる。物そのものに価値がある。

2　資源の枯渇・廃棄物処理問題が生じる恐れがある。

3　製品毎に別梱包され、購入者への到達経路も様々である。

4　盗難に対しては、物理的・法律的防御がなされている。

ーーーデイジタル情報ーーー

1　Ａ者からＢ者へはコピーを渡せばよく、自己の持つ財物は、なくならない。

2　資源の枯渇・廃棄物処理問題は、情報自体には生じないが、使用媒体によっては生じることがある。

3　規格化されたインデックスの存在により製品毎に別梱包される必要はなくなる。
購入者への到達経路も、一様でよい。

4　盗難状況が把握し辛い。

６．超流通の仕組み

超流通では、情報製品を一般には所有するのではなく、使用記録をとり、その記録に応じて、料金の支払をすることになる。使用記録は、勝手に改変されぬよう保護されている。

以下に仕組みを、利用者と提供者の立場にたってのべる

ーーー利用者ーーー

1　入手方法

ネットワークを介する
ＣＤ等の大容量媒体を安価あるいは無料で入手する

2　記録方法

専用のＩＣを手持ちのコンピュータに設置してメモリカードに書き込む

3　回収方法

メモリーカードを読み取り機のある店舗に持参する
電話回線・専用線を通して電子的に吸い上げる。

4　支払方法

クレジット会社と契約の上、支払。
記録回収機の設置店舗での直接支払。

ーーー提供者（著作権者・超流通サービス提供者）ーーー

1　権利の登録

提供者は、始めに超流通システム取り扱い機関との間で、契約を結び、自己のデイジタル情報に関する権利を登録する。

2　使用料の受取

上記、契約内容に従い、料金の分配を受ける。

7．超流通での課金方法

超流通のための料金システムは、
従来の買取料金のほかに、従量料金・無料使用・試用料金等の
「標準の料金体系」と、

使用者の所属・社会的行事・出来事等によって料金を変える
ことのできる
「特別許諾による料金体系」

の２つに大別されている。

標準の料金体系

1　試用課金

提供者が設定した限度まで無料または低料金で試しに使うことができる

2　従量課金

使用回数・使用時間・個別サービス（印刷やデータ保存）を基準に課金される。
料金は提供者が自由に設定できる。

3　自動買取

使用累計額が提供者によって定められた上限を越えるか、又は指定した日時を過ぎると、それ以降は課金されない。

4　無料使用

但し使用状況報告の義務はある。

特別許諾による料金体系

5　特別許諾

特別許諾条件の設定により、教育機関割引、集団契約割引、機関限定割引等の提供ができる。

8．超流通での社会

超流通は、

1　電子的に添付された使用条件

2　電子的に回収・処理される使用記録

を、構成要素とした、デイジタル情報の流通・供給システムである。流通システムが社会的に成立するためには、基本的に利用者と、提供者との、合意が必須であるが、超流通では、双方にとって無駄を省き、コピーを認めた画期的システムといえる。

そして、

価値あるデイジタル情報を生産・加工したものは誰でも、安価でシンプルな流通システムにより、大量に情報を提供できる。

また、

流通範囲は、広く世界中のあらゆる情報を迅速に・一様に利用できる。

その結果

ソフトウエアの生産に飛躍的な発展・変化をもたらすと共に、ソフトウエアの部品の市場を形成するようになる。

ダイヤルQ^2を利用したソフトウェア流通の可能性について

日本電信電話株式会社　北海道総合通信システム本部

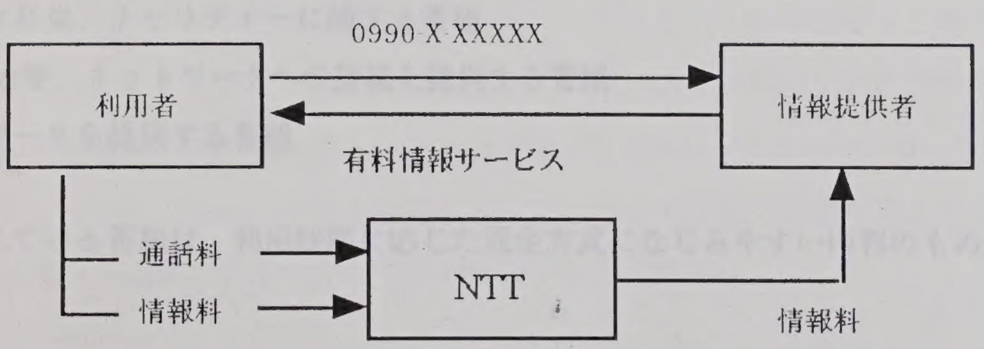
中島孝志

1　サービスの概要

1-1　ダイヤルQ^2サービスとは

ダイヤルQ^2とは、NTTが提供するサービスであり、IP（Information-Provider：情報提供者）が提供した情報に対する情報料を、NTTがIPに代行して回収するサービスである。情報料は、毎月の電話料金と一緒にダイヤルQ^2利用料として請求され、利用者は、クレジットカード番号を通知するなど、IPと別個の申込契約、支払手続きをとる必要がなく気軽に利用できるメリットがある。またIPにとっても、NTTが回収代行を行うことによって料金回収等の業務から解放されるメリットを持つ（図-1）

（図-1）ダイヤルQ^2の仕組み


0990-X-XXXXX
利用者
情報提供者
有料情報サービス
通話料
情報料
NTT
情報料

1-2 ダイヤルQ^{2}サービスの課金方式

ダイヤルQ^{2}では、利用者に通常の通話料金と一緒に情報料が請求される。NTTの仕様では、おもに音声系の情報提供サービスでの利用を意識しているため、接続時間単位による課金方式のみの提供となっている。（表-1）

（表-1）

番号帯	3分間あたりの情報料
0990-6-XXXXX	10円～150円 （7段階）
0990-5-XXXXX	180円～300円（5段階）
0990-3-XXXXX	10円～300円 （12段階）

注）0990-3-XXXXXの番号帯で提供する番組のうち、パーティーラインについては
　上限が60円まで。

1-3 ダイヤルQ^{2}サービスの利用の現状

ダイヤルQ^{2}サービスの利用の現状としては、下記のような内容の番組が挙げられる。

①ニュース、経済情報、株式情報を提供する番組
②教育、教養に関する情報を提供する番組
③法律、生活相談に関する情報を提供する番組
④特定利用者間の通信を媒介する番組（伝言型番組等）
⑤成人向け番組
⑥不特定利用者間の通信を媒介する番組（パーティーライン等）
⑦義援金募集、チャリティーに関する番組
⑧Internet等、ネットワークへの接続を提供する番組
⑨各種データを提供する番組

提供されている番組は、利用時間に応じた課金方式になじみやすい内容のものが多くみられる。

2　ソフトウェアを流通させるうえで必要とされる機能仕様

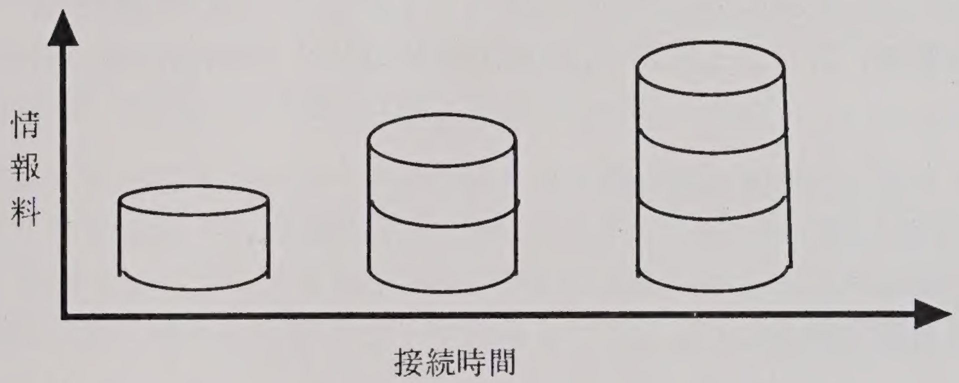
2-1　新たな課金方式

ダイヤルQ^{2}を利用してソフトウェアを流通させるうえで必要となるのは、音声系以外の情報を流通可能とさせる課金方式であろう。

現状のダイヤルQ^{2}サービスの仕様では、ソフトウェアを流通させる場合、コストを決定する要因として重要な意味を占めるはソフトウェアのデータ容量の問題である。現状では、接続時間に比例した課金となっているため、データ容量の大きなソフトウェアを伝送するということは、より長い接続時間、より高い課金となるわけである。（図-2）

反面、ソフトウェア自体の持つ価格（価値）は、必ずしもデータ量と比例するわけではない。ソフトウェアをネットワーク上で流通させるにあたっては、これらの背景をふまえた新たな課金方式の設定が重要となることは言うまでもない。ダイヤルQ^{2}サービスを利用しソフトウェアを流通させるには、接続時間によらず、流通させるコンテンツに応じて課金設定を行うという、新たな課金方式が必要である。

（図-2）現状のダイヤルQ^{2}の課金



2-2　回線の伝送速度

ソフトウェアのデータサイズは年々大容量化しているため、利用者がネットワークを通じてソフトウェアを入手しようとするとき、ソフトウェアの伝送速度に起因する伝送時間の長時間化すなわち回線コストの増加の問題が生じる。これはダイヤルQ^{2}サービスだけに限った問題ではないが、将来的に、ネットワークサービスを利用したソフトウェア流通の問題を考えるとき、高速な伝送速度を持つ回線を低価格で利用できる環境が必要となるであろう。

3　ダイヤルQ^{2}を利用した情報提供サービスの今後の可能性

3-1　コンテンツビジネスへの応用

2-1で述べた接続時間によらない新たな課金方式が提供されることにより、これまで課金面から対応できなかった新しいタイプのサービスへの応用が可能と思われる。

①ソフトウェア、シリアルキー等の流通、販売
②テキスト、動画像、音声データ等の各種著作物の流通、販売
③決済機能のみを提供するサービス

今後の情報のディジタル化、マルチメディア化、ネットワークを利用した流通という動向を考えるとき、データそのものの形態（データ容量、データ形式）から表現することが困難な、情報そのものの持つ価値を、いかに表現していくかということがますます重要になってくであろう。ネットワーク上での課金という問題は、そういった意味からもより一層重要なテーマとしてとらえていかなければならない。

Survey of Electronic Payment System
電子決済の仕組み

大久保清貴

1995年3月22日

概要

3000万人いるといわれるインターネットのユーザをビジネスのターゲットとする動きが急である。特に情報を商品としてインターネット上でユーザと直接取引きするための支払方法や通貨がホットなテーマになっている。本文では、特にアメリカで起こっている動きをサーベイする。

1　インターネット上でのビジネス

インターネットはネットワークをつなぐネットワークとして拡大してきた。90年代に入ってアメリカ政府の情報スーパーハイウェイ、NREN(National Research and Education Network)、NII(National Information Infrastructure) [1][2] 構想とともに、商用化が進み急速な発展をとげている (日本版 NII[3]、日本版 GII[4], ヨーロッパ版 GII[5])。

インターネット上のネットワークならびホスト数の統計資料によると [6]、1995年1月現在、直接接続されているホスト数は約485万台[1]である。1994年の第4四半期に26% 増加している。なかでも商用ネットワークを表す com の増加が著しい。ちなみに日本は1994年中に86%（第4四半期は17%）増加して約9万7千台になった。図1は成長が加速している様子を示している [7]。

ユーザ数については、統計方法がなく推定するしかないが、2000万人とも3000万人とも言われている。「2003年までには地球上の全人類がつながる。[23]」という説もある。

ここ数年でインターネットが爆発的に成長した要因は、WAIS、gopher、World Wide Web(WWW) といっただれにでも簡単に情報にアクセスできるブラウザが開発されたことにある。94年7月に東京で開かれた Networld+Interop94 で ISOC(Internet Society) の Anthony M. Rutkowski 氏が語ったところによると [8]、「WWW のようなアプリケーションが今の勢いで伸びると、3年後には電話回線上の声をデジタル化した通信量をインターネットの通信量が上回るであろう」と予測されている。図2は一時勢いのあった gopher を WWW が抜きさる様子を表している [9]。

これまでインターネットに関するビジネスとしては、バックボーンとなる長距離網の接続サービ

---

[1] firewall の後ろにあったりドメイン名が登録されていないなどの理由でアクセスできないものは含まれていない。使用時だけアクセス・プロバイダに PPP や SLIP で接続するものも含まれない。

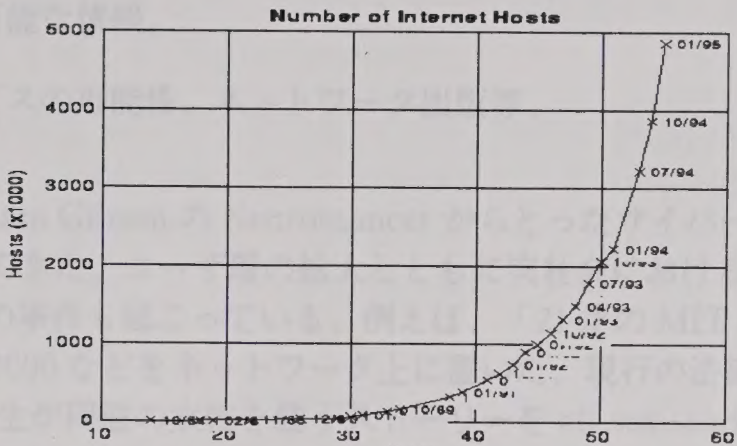


図 1: インターネット上のホスト数の増加

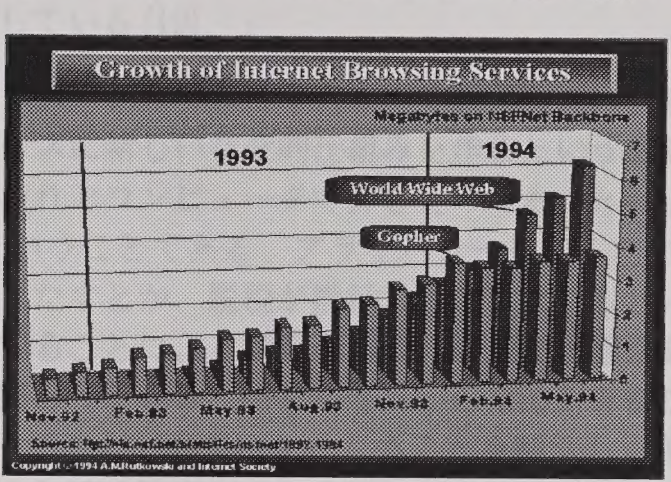


図 2: Internet 上でのブラウジング・サービスの増加

ス[2]、地域網とユーザの接続サービス[3]、パソコン通信からアクセスできるサービス[4]の提供が主であった。最近は Mosaic や Netscape などマルチメディアを扱える WWW ブラウザの浸透にともない、商店の店先 (store front) を広告としてだして (network presence) バーチャル商店街を構成するケースが急増している。これらの店の中にはクレジット・カードで商品を買えるものも多い。その中には現実に売っているものをそのままもちこんだケースも多くみられるが、雑誌、新聞、データベース、コンサルタント等の情報そのものを商品とする場合もあり、このときはネットワーク上ですべての取引きを完結できるという利点がある。

インターネットをビジネス・チャンスとみなす要因として次の点をあげることができる。

- 3000 万人ともいわれるユーザ。
- 比較的偏ったユーザ層 [10]。男子学生、技術系、共和党支持者が多いなど。
- ニュース・グループからセグメント化されたユーザ層をピックアップできる。
- オープンな競争。入札など。

---

[2] AT&T、MCI 等

[3] Access Provider または Internet Service Provider

[4] CompuServe, AOL, Prodigy 等

- 企業間の協力。
- 迅速な市場把握、製品投入
- 膨大なアクセス可能な情報。
- 新しい情報サービスの可能性。ネットワーク出版等。

インターネットを William Gibson の Neuromancer からとったサイバースペース (cyberspace) と表現することが定着してきた。ユーザ層の拡大とともに実社会における事件に対応することや、ネットワークならではの事件も起こっている。例えば、「21 才の MIT の学生が Execl5.0、 WordPerfect6.0、 Sim City 2000 などをネットワーク上に置いた。現行の法律では厳しく罰せない [17]。」、「20 才のミシガン大学生が同窓の女性を襲うストーリーを alt.sex.stories に流したために逮捕された。この女性も仕返しのストーリーをポストした [18]。これに対して、上院議員がネットワーク上にワイセツな文書等を流してはならないという法律の提出したが、EFF などが言論の自由への抑圧につながると反対している [19]。」

インターネット上でのビジネス・システムを考える上で重要なのは、単にデジタル化うんぬんというよりは、サイバースペースに新しい社会的仕組みを作るという感覚をもつことである。よく言われるのは、金や情報の流れをコンピュータで押さえられることによってプライバシーが守られなくなるのではないかということである。プライバシーを守るために暗号を使えばよいかというと、犯罪防止に支障をきたすとか、脱税が横行することになるという意見もある。また国境を越えて情報が自由に流通することによる問題もある。さらに情報を商品とする場合は、表現の自由、著作権の問題が深く関わっている。たとえば NII の知的所有権ワーキンググループの提案 [20] では、デジタル化によってコピーが自由に作れることを懸念して著者や出版社の権限を拡大するようもとめているが、これでは読者がこれまでもっていた、本屋でブラウジングしたり本を友人に貸すこともできなくなるという声もある。

インターネット上でもっとも有望視されているビジネスのひとつにネットワーク出版がある。WWW がそうであるように、ネットワークをあたかも巨大なハイパー・ドキュメント・ベースのようにしてナビゲーションすることができる。ビジネス化するとき重要なのは、他の著者の文書を（自由に）引用／参照でき[5]、読んだページに応じて料金が各著者に払われる仕組みである。このことは Ted Nelson 氏が Xanadu[6]として 1960 年代から考えていたことである [21][22]。この非常に小額の支払い（マイクロ・ペイメント）ができるかどうかが、電子決済システムの適用範囲を大きく左右する。

2 暗号

暗号は秘密で通信を行う為に、書き方の発明と同じくらい古くから用いられている [11]。最近の暗号の発展で重要なのはコンピュータの発明と公開キー暗号方式 (public key encryption system) である。公開キー暗号方式が、電子通貨を可能にしたといっても過言ではない。これは 1976 年に Diffie と Hellman によって発明された [12]。それ以前の暗号は秘密キー暗号方式 (secret key encryption system) といって送り手が暗号化するのに使ったのと同じキーを用いて受け手は解読した (symmetric key)。このため受け手にキーを送るのがたいへんだった。それに対し公開キー

---

[5] 著作権の新しい仕組みが必要となる。

[6] Xanadu は Project Xanadu の登録商標である。

暗号方式では、送り手が暗号化のために使うキーとが受け手が解読するキーが別になっている(asymmetric key)。そこで送り手が使うキーを公開しておき、受け手のキーを秘密にしておけば、特定の受け手だけが解読できるメッセージを送ることができる。逆に秘密のキーを使って暗号化したメッセージはだれにでも解読できるが、このメッセージは秘密のキーを知っている人だけしか作成できないので署名として使える(digital signature)。

代表的な公開キー暗号方式はRSAである[13]。RSAの暗号の解読の難しさは、数値を素数分解する難しさからきている。RSAで暗号化したり解読したりするのは時間がかかるので、RSAは秘密キー暗号方式のキーを送るために使うことが多い。もしメッセージがいじられなかったことを保証するだけでよいばあいは、メッセージを送る際にチェックサム(message digest)をとり、その部分だけを暗号化すればよい。

ここ数年、アメリカでClipperという通信の暗号を行うチップが議論をよんでいる[14]。もともとFBIの要求に応えてNSA(National Security Agency)が作ったものである。NSAは政府の暗号に関する必要性(海外駐在員との秘密の通信、外国の秘密の通信の解読等)を満たしてきた組織である。近年携帯電話等の普及により盗聴の問題が顕在化してきた。通話を暗号化すればよいのだが、RSA等の破るのが困難な暗号が使われると、FBIがこれまで裁判所の承認を得て行ってきた犯罪摘発等のための盗聴ができなくなってしまう恐れがある。この矛盾した問題を解決するために考え出されたのが、第三者(政府機関)にキーを預ける方式(Escrowed Encryption System)である。これは、通話は暗号化されるのでプライバシーがまもられるが、いったん預けたキーを用いればたやすく盗聴できるものである。

PGP(Pretty Good Privacy)[15]は、Phil Zimmermannがつくった暗号ツールであり、RSA等のアルゴリズムを含んでいる。PGPはインターネット上に置かれたためにたちまち広がってインターネット上の暗号ツールの事実上の標準になった。暗号には各国ごとに異なる規制が定められている。例えばアメリカでは暗号用のソフトとハードは兵器と見なされている。フランスでは届けられた方式以外の暗号文は通信回線上をながしてはならないことになっている。インターネットには国境がないのでPGPを使って暗号化したメールを送ることや、海外のサイトにあるPGPのソフトを取ってくることを規制する有効な手段がない。現に民主化のためのゲリラ戦にPGPをつかっている国もある。

3　ビジネスの動き

3.1　CommerceNet

CommerceNetは、1994年から始まった北カルフォルニアの地域情報インフラ(regional information infrastructure)を構築するためのコンソーシャムであり、BBN BARRNET, Enterprise Integration Technology(EIT), スタンフォード大学のCenter for Information Technology(CIT)が中心となって作った非営利団体である。シリコンバレーの復権を狙うベンチャー企業Smart Valley社、カルフォルニア州もスポンサーになっている。コンソーシャムのメンバーはシリコンバレー近辺の主要な企業を含む約120社にのぼる。

CommerceNetは、「主要な企業、組織は5年以内にインターネットでビジネスを始める。」としており、他の電子取引システム(electronic commerce)とともに事実上のNIIへの第一歩を目指している。目的のひとつとして、、専用回線からインターネットへ乗り換えることで電子取引システムへの参入に要する時間とコストを下げることをかかげている。具体的には、デイレクト

暗号方式では、送り手が暗号化のために使うキーと受け手が解読するキーが別になっている (asymmetric key)。そこで送り手が使うキーを公開しておき、受け手のキーを秘密にしておけば、特定の受け手だけが解読できるメッセージを送ることができる。逆に秘密のキーを使って暗号化したメッセージはだれにでも解読できるが、このメッセージは秘密のキーを知っている人だけしか作成できないので署名として使える (digital signature)。

代表的な公開キー暗号方式は RSA である [13]。RSA の暗号の解読の難しさは、数値を素数分解する難しさからきている。RSA で暗号化したり解読したりするのは時間がかかるので、RSA は秘密キー暗号方式のキーを送るために使うことが多い。もしメッセージがいじられなかったことを保証するだけでよいばあいは、メッセージを送る際にチェックサム (message digest) をとり、その部分だけを暗号化すればよい。

ここ数年、アメリカで Clipper という通信の暗号を行うチップが議論をよんでいる [14]。もともと FBI の要求に応えて NSA(National Security Agency) が作ったものである。NSA は政府の暗号に関する必要性 (海外駐在員との秘密の通信、外国の政府の通信の解読等) を満たしてきた組織である。近年携帯電話等の普及により盗聴の問題が顕在化してきた。通話を暗号化すればよいのだが、RSA 等の解るのが困難な暗号が使われると、FBI がこれまで裁判所の承認を得て行ってきた犯罪捜査等のための盗聴ができなくなってしまう恐れがある。この矛盾した問題を解決するために考え出されたのが、第三者 (政府機関) にキーを預ける方式 (Escrowed Encryption System) である。これは、通話は暗号化されるのでプライバシーがまもられるが、いったん預けたキーを用いればたやすく盗聴できるものである。

PGP(Pretty Good Privacy)[15] は、Phil Zimmermann がつくった暗号ツールであり、RSA 等のアルゴリズムを含んでいる。PGP はインターネット上に置かれたためにたちまち広がってインターネット上の暗号ツールの事実上の標準になった。暗号には各国ごとに異なる規制が定められている。例えばアメリカでは暗号用のソフトとハードは武器と見なされている。フランスでは届け出られた方式以外の暗号文は通信回線上をながしてはならないことになっている。インターネットには国境がないので PGP を使って暗号化したメールを送ることや、海外のサイトにある PGP のソフトを取ってくることを規制する有効な手段がない。現に民主化のためのゲリラ戦に PGP をつかっている国もある。

3　ビジネスの動き

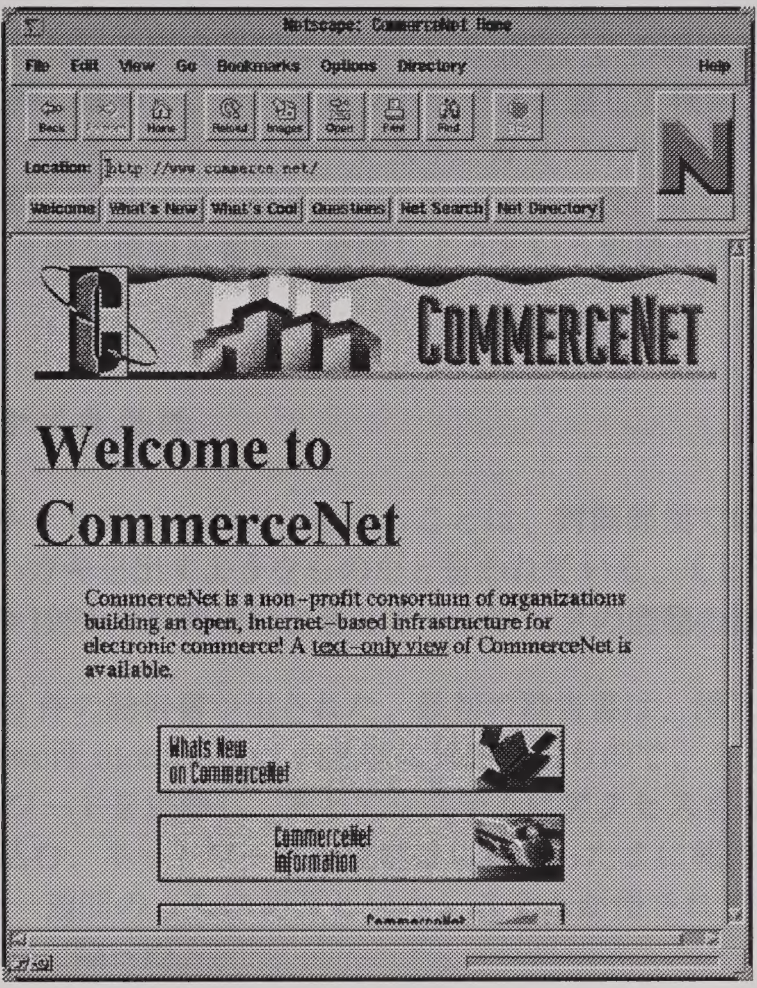
3.1　CommerceNet

CommerceNet は、1994 年から始まった北カルフォルニアの地域情報インフラ (regional information infrastructure) を構築するためのコンソーシャムであり、BBN BARRNET,Enterprise Integration Technology(EIT), スタンフォード大学の Center for Information Technology(CIT) が中心となって作った非営利団体である。シリコンバレーの底流を担うベンチャー企業 Smart Valley 社、カルフォルニア州もスポンサーになっている。コンソーシャムのメンバーはシリコンバレー近辺の主要な企業を含む約 120 社にのぼる。

CommerceNet は、「主要な企業、組織は 5 年以内にインターネット上でビジネスを始める」としており、他の電子取引システム (electronic commerce) とともに事実上の NII への第一歩を目指している。目的のひとつとして、、専用回線からインターネットへ乗り換えることで電子取引システムへの導入に要する時間とコストを下げることをかかげている。具体的には、ディレクトリ

リ・サービス、仲介・照会、業者の信頼度レポート、金融・運送サービス、情報蓄積を整備することになっている。

CommerceNet は、EIT 社と RSA Data Security 社のジョイント・ベンチャー会社 Terisa Systems が開発した Secure HTTP を使う。


Netscape: CommerceNet Home
File
Edit
View
Go
Bookmarks
Options
Directory
Help
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Location: http://www.commerce.net/
Welcome
What's New
What's Cool
Questions
Net Search
Net Directory
COMMERCENET
Whats New on CommerceNet
CommerceNet Information
CommerceNet

Welcome to CommerceNet

CommerceNet is a non-profit consortium of organizations building an open, Internet-based infrastructure for electronic commerce! A text-only view of CommerceNet is available.

図 3: CommerceNet のホームページ

3.2 Store Front

Web を使った広告／店先／カタログを提供しているサイトは山ほどあるが、そのいくつかを下にあげる。

MarketNet[24]、MarketPlace.com[25]、DealerNet[26]、Internet Business Directory[27]、Shopping2000 Home Page[29]、NetMarket[30]、CyberMalls[31]、Internet Mall[32] First Virtual[46] Global News Navigator[33]

3.3 InfoMall

InfoMall は NII の研究の中核をなす High Performance Computing and Communication(HPCC) プログラム[7]で開発された高度技術を移転し HPCC ソフトウェア産業を起こすことをねらっている [34]。このモールにはテナントがいて顧客がやってきてこのマーケットで何が起きているかを

---

[7] 並列コンピュータやギガビット・ネットワークを研究している。

知ることができる。CommerceNet が地域主導の NII へのアプローチなのに対して、InfoMall は政府の後押しがあるアプローチと言えよう。

3.4 デジタル・ライブラリー

デジタル・ライブラリーも NII の研究課題に入っている [35]。この研究では、情報の収集、蓄積、体系化をドラマチックに押し進めて、それをネットワークを通して検索、処理することを目指している。カーネギーメロン大学、スタンフォード大学、カリフォルニア・バークレー校、カルフォルニア・サンタバーバラ校、イリノイ・ウルバナキャンペィン校、ミシガン大学が参加している。情報に課金できないと、有用な情報を蓄積、公開しようという動機が得られないという認識がある。

4 電子決済が満たすべき条件

セキュリティ インターネットでメッセージを送ることは、封筒郵便ではなく、途中で誰に読まれるかわからないハガキを出すようなものだと言われている。CERT(Comupter Emergency Response Team) の 1994 年度報告によるとネットワークの盗聴の被害が急増している [49]。

> 1994 年には 15 件の勧告書が出された。最初の勧告書は、侵入者によるネットワークの盗聴 (network monitoring) の急増に対応するもので、2 月 3 日に出された。侵入者はパスワードを盗聴し、あとでシステムにアクセスするために裏戸 (back door) を組み込む。ネットワークの盗聴は以前からたまに報告されることがあったが、たった一週間のうちに何万というシステムが侵されてしまったかもしれないという事態に陥った。

規模拡大性 (scalability) 中央でデータベースに集中管理しなければならないようでは、インターネットの規模をカバーできない。

匿名性 (anonymity) 普通の紙幣では、どこから入手してどこで使ったかという記録が残らない。もし記録から誰が何を買ったかすぐわかる方式だとプライバシーが守られなくなる。

流通性 だれもが価値を認めるものでなければ金としての意味が低い。

効率 処理にコストがかかり過ぎては使いものにならない。

組み込み易さ アプリケーションに簡単に組み込めて、ユーザが細部を気にしなくても使えるようにする必要がある。

扱い易さ 電子貨幣では、携帯性があったり、紛失等の事故に簡単に対処できる必要がある。

5 電子決済の具体例

5.1 Mondex

Mondex[43] は、イギリスのスインドン (Swindon) で 95 年から実験が開始される予定のスマートカードである。Mondex 対応の公衆電話、電子財布、携帯電話、PDA (personal digital assis-

tant) を通して金のやり取りができる。銀行のコンピュータで集中管理する必要がない。取引きの身元確認にデジタル署名を使う。当初、データ自体は暗号化しないが、必要に応じて対応できる。

スマートカードが受け入れられるためには、もらった電子通貨がちゃんと銀行で換金できるものでなければならない。ハッカーが偽造通貨を発行して通貨供給量を一挙に増やして経済が破綻するようでは困る。

5.2 First Virtual

First Virtual Holding 社 [44] は情報をインターネット上で情報の売り買いを仲介する。この方法は、もっとも簡単であり、暗号は使わないし、インターネット上をクレジットカードの番号が流れることもない。

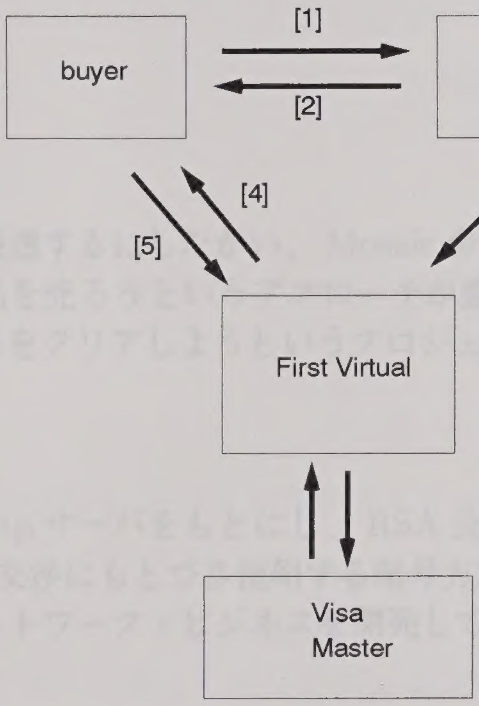


図 4: First Virtual のプロトコル

買い手 (buyer) と売り手 (seller) はあらかじめ First Virtual に名前、電子メールアドレス、カード番号を教えて、アカウントを作ってもらい、アカウント番号をもらう。情報を買うときはアカウント番号で行う。プロセスは次のようになる (図 4参照)。

1. 買い手は、欲しい情報を持っている売り手にアカウント番号を使って注文する。
2. 売り手は情報を買い手に送る。
3. 売り手は情報を誰に送ったかを First Virtual に知らせる。
4. First Virtual は買い手に、送られてきた情報に満足したかを尋ねる電子メールを出す。
5. 買い手は満足していれば Yes を、不満なら No を、注文した覚えがなければ Fraud を First Virtual に知らせる。 Fraud の場合は、 First Virtual のアカウント番号を変更してもらう。

First Virtual のアカウントにある程度の金額がたまったところでカード会社にクレジットの要求を出す。こうすることで、コストのかかるカードでの決済の回数を減らすことで小額取引きを可能としている。

この方法では、情報を送っても支払が行われない場合もあるので、売り手のリスクが大きい。買い手にとっては、支払を行う前に情報を見て有用かどうか判断できるのでありがたい。

現在、 CompuServe や Prodigy で行われているのは、買い手がすでに売り手への支払にクレジット・カードを利用できるようにしている場合には、オンラインで決済が可能というものである。これに対し、First Virtual の方式を用いれば、買い手と売り手があらかじめ関係をもっていなくても、オンライン決済ができる。

電子決済の仲介業を行おうとするいずれの会社にも言えることだが、クレジットカード番号や個人情報が集まるので、会社への信頼をもってもらう必要性が高い。このような会社は、金融関係で実績をもった人物とコンピュータのセキュリティで名の通った人物などが集まって作られていることをアピールしている。

5.3 Secure HTTP

World Wide Web が一般に浸透するにしたがい、 Mosaic や Netscape を通して、クレジットカードを使って直接買い手に商品を売ろうというアプローチが盛んである。そのばあい問題になるのがセキュリティであり、これをクリアしようというプロジェクトがいくつかある。

- Terisa Systems [36]

  NCSA の Mosaic と http サーバをもとにし、RSA 公開暗号方式を組み込んでいる。クライエント／サーバ間の交渉にもとづき使用する暗号方式を変えられる。Open Market 社、CyberCash 社などネットワーク・ビジネスを開発している企業が採用。

- Spyglass [37]

  拡張 Mosaic ブラウザ。マイクロソフト社、 AT&T、 DEC、 IBM、 NEC、 NTT、 PSI などが Spyglass のライセンスを受けことになっている。

- Netscape Communications Corp. [39]

  クライエントの Netscape Navigator とサーバの Netsite Communication Server が Secure Socket Layer(SSL) というセキュアなチャネルを使って通信する。電子ショッピング・システム InterMCI がこれを使う予定になっている。

- IETF HTTP Security Working Group [42]

  インターネットの標準をつくる Internet Engineering Task Forc(IETF) で検討中。

5.4 NetBill

カーネギーメロン大学 infomedia Digital Library プロジェクトで開発し、Visa[47] も参加して 1995 年後半から実験予定。

NetBillは、カーネギーメロン大学で開発された、売り手(merchant)と買い手(consumer)を結ぶオープンな標準に基づく電子取引システムである。このシステムは、例えば1ページの文書を10セントで売ったときに、1セントの手数料ですむようにすることを狙っている。また、情報を顧客に正しく配送できたときにだけ支払が行われるような仕組み(certified delivery mechanism)をもっていることが特徴である。

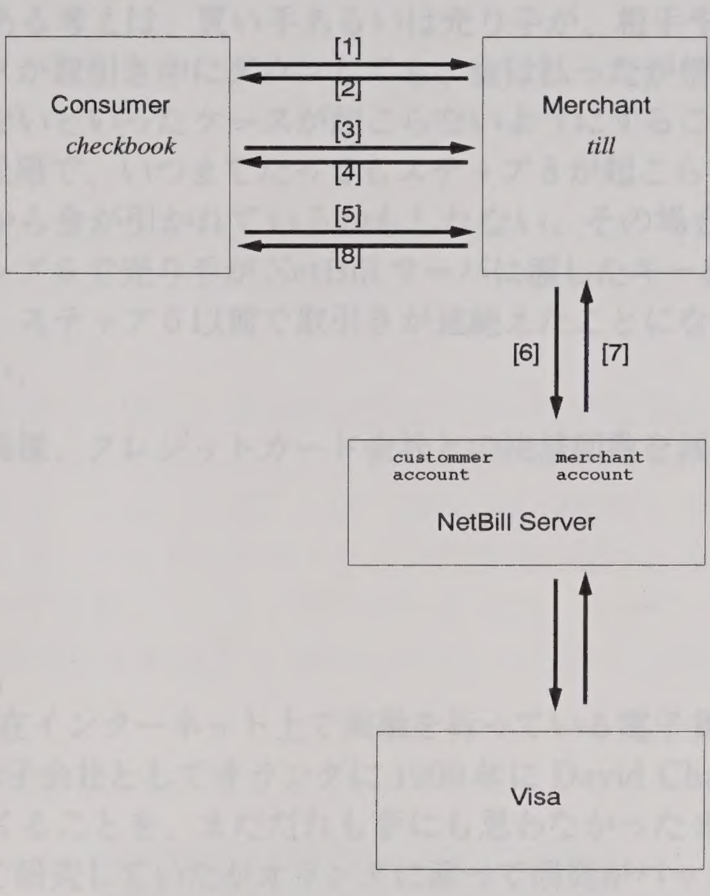


図5: NetBillにおけるトランザクションのプロトコル

NetBillのトランザクション・プロトコルを示す(図5参照)。あらかじめNetBillサーバに買い手と売り手はアカウントを作っておく。買い手側のアプリケーションは取引きのための処理をおこなうcheckbookというライブラリを通して売り手側と通信する。同様に売り手側もtillというライブラリを通して買い手側と通信する。

このプロトコルで使われるメッセージはすべてデジタル署名を付けて送られる。このため、メッセージの発信者は、メッセージを発信したことを否定できない。

1. 買い手側のアプリケーションがcheckbookを通して、売り手側のtillに価格を尋ねる。
2. tillが価格を計算してcheckbookに価格を返す。
3. 値段が適当ならcheckbookは購入要求をtillに送る。
4. tillはアプリケーションから情報をもらい、それを暗号化してcheckbookに送る。暗号を解く鍵はこの時点では送らない。情報が途中で壊れなかったことを調べるためにチェックサムをとっておく。
5. checkbookは送られてきた情報のチェックサムと購入価格などをtillに返す。
6. tillは送られてきたチェクサムと以前に自分で計算しておいたチェックサムを比較し、違っていれば情報を再送する。売り手側のアプリケーションは送り状と暗号を解くキーをNetBillサーバに送る。

7. NetBill サーバは買い手のアカウントから売り手のアカウントへ金を動かす。NetBill サーバは正常に送金ができたかどうか till に返す。

8. 正常に送金できた場合は、暗号を解くキーを checkbook に送る。

このプロトコルの中心にある考えは、買い手あるいは売り手が、相手や NetBill サーバをだませないことと、ネットワークが取引き中にダウンしても、金は払ったが情報をもらえないとか、情報は送ったが金がもらえないといったケースが起こらないようにすることである。例えば買い手がステップ 5 まで進んだ段階で、いつまでたってもステップ 8 が起こらない場合は、すでに NetBill サーバのアカウントから金が引かれているかもしれない。その場合は、買い手側は NetBill サーバと通信して、ステップ 6 で売り手が NetBill サーバに渡したキーをもらう。もし NetBill サーバがキーを知らなければ、ステップ 6 以前で取引きが途絶えたことになるので、取引き自体を中止してもだれも損をしない。

NetBill も First Virtual 同様、クレジットカード会社との決済回数を減らすことで小額取引きを可能としている。

5.5 ecash

ecash は DigiCash 社が現在インターネット上で実験を行っている電子貨幣である。DigiCash 社はアメリカ DigiCash 社の子会社としてオランダに 1990 年に David Chaum により設立された。彼は、電子貨幣の時代がくることを、まだだれも夢にも思わなかった 80 年代半ばから研究してきた。もともとアメリカで研究していたがオランダに渡って研究がバックアップされた [58]。digicash は、「数字が金」[57] であり、その特徴は

- ブラインド署名 (blind signature) により、使用者のプライバシーが守られる (unconditionally untraceable electronic cash)。証明に必要な最小限の情報だけを相手に伝える。
- 同じ電子貨幣を複数回使うと、使った人の身元がばれる。一回の使用では身元の追跡ができないが、何回か使うと身元が判明する。

同じ電子貨幣を複数回使うと身元が判明する仕組みを理解するには、Adi Shamir の論文 [61] にある「データ D を n 個に分けて、そのうち k-1 個ではデータ D が絶対再現できないが k 個あれば再現できる」にはどうすればよいかという問題を考えてみればよい。

簡単に電子通貨のモデルを考えてみよう。徐々に複雑だが実用的なモデルになる。

1. 預金額と引き替えに銀行が紙幣番号を発行する。

   すべての人間が、かってに通貨番号を発行したり 2 回以上つかったりしないほど正直でミスもしなければ、これでもよい。

2. 預金額と引き替えに銀行が通貨番号にデジタル署名をして発行する。

   かってに通貨番号を発行されることは防げる。

3. 預金額と引き替えに、預金者がデジタル署名した小切手番号に銀行が通貨番号をふり、これにデジタル署名をして発行する。

   銀行は、同じ通貨が 2 回以上使われた場合に発見できるよう、預金者と通貨番号の対応をデータベースとして管理するので、売り手がこの通貨を銀行に預金したときに通貨がどの買い手との取引きにつかわれたかが分かってしまう。

4. 預金額と引き替えに、預金者が小切手番号を一方向関数で変換したものにデジタル署名したものに銀行が通貨番号をふり、これにデジタル署名をして発行する。

digicash の方式は、暗号の仕組み自体によって信頼性を高めている。これにたいし Mondex の方式はスマートカードがいじられない (tamper-free) ことで信頼性を高めようとしている。このことは digicash 方式の強味である。

digicash の方式では電子通貨が 2 回以上使われたかどうかは銀行に預金されている通貨番号が再び預金されることがあるかどうかで調べる。したがって金額が大きい場合はオンラインで取引き時にチェックすることになる。しかしこのことは、銀行が集中管理している通貨番号との比較を伴うのでコストがかかる。そこでなるべくオフラインで通貨のやりとりができるよう信頼性の高いスマートカードもあわせて提案している。このカードには、ユーザ・サイドを代表する representative と、公的機関が信頼性を保証する observer の 2 つのパーツが併存する。これらパーツは互いを監視しあう。

図 6に DigiCash 社が現在実験中の電子通貨 ecash のクライエントを示す。

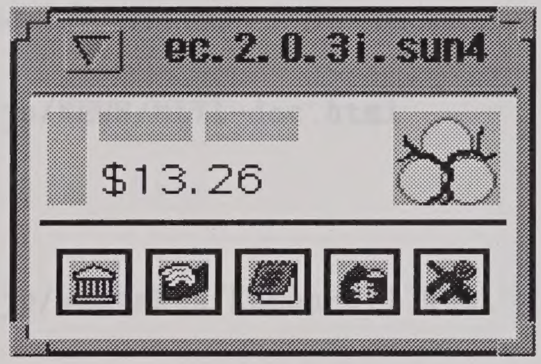


図 6: X ウィンドウで動く ecash クライアント

Chaum の書いた論文をもとに、 digicash 方式を実験しているグループが DigiCash 社とは別にある。 Pr0duct Cypher というグループが実験のために Magic Money というシステムを作っている。 Magic Money を扱う Phantom Exchange という銀行もある。 Community ConneXion というバークレー大学のグループは NexusBucks という Magic Money をベースにしたシステムを作っている。

5.6　その他

その他にも以下の様なさまざまな試みがされている。

- Brands [62]
- LETSystems [64]

- AT&T Bell Lab Buyinfo [65]
- NetCash [68]
- NetCheque [69]
- NetChex [70]
- CyberCash [73]
- NetCheque [69]
- XIWT [74]

参考文献

[1] *Index of /nii/*

`http://sunsite.unc.edu:80/nii/`

[2] *National Information Infrastructure: Table of Contents*

`http://www.hpcc.gov/blue95/index.html`

[3] *MITI Program for Advanced Information Infrastructure*

`http://www.glocom.ac.jp/NEWS/MITI-doc.html`

[4] *Kumon on GII*

`http://www.glocom.ac.jp/WhatsNew/Kumon-GII.html`

[5] *Europe and the global information society - Bangemann report*

`http://www.earn.net/EC/bangemann.html`

[6] *Internet Domain Survey*

`http://www.nw.com/zone/WWW/top.html`

[7] *Number of Internet Hosts*

`http://www.nw.com/zone/hosts-graph-linear.gif`

[8] *Rutkowski Keynote: The Present and the Future of the Internet*

`http://info.isoc.org/interop-tokyo.html`

[9] *Growth of Internet Browsing Service*

ftp://ftp.isoc.org/isoc/charts/traffic-gifs/www-byte.gif

[10] *GVU Center's 2nd WWW User Survey Home Page*

http://www.cc.gatech.edu/gvu/user_surveys/survey-09-1994

[11] *The Five Great Inventions of Twentieth Century Cryptography*

http://www.wired.com/Lib/Privacy/crypto.inventions.html

[12] *"Prophet of Privacy" by Steven Levy*

http://www.wired.com/Lib/Wired/2.11/features/diffie.html

[13] *RSA Data Security, Inc.'s Home Page*

http://www.rsa.com/

[14] *EFF Clipper/EES/Capstone/Tessera/Key Escrow Archive*

http://www.eff.org/pub/EFF/Policy/Crypto/Clipper/

[15] *PGP 2.6.2 hypertext documentation*

http://www.pegasus.esprit.ec.org/people/arne/pgp.html

[16] *GNN I-Media Center: Links to Mitnic*

http://gnn.com/gnn/meta/imedia/resources/mitnick.html

[17] *CyberLaw (tm) 1/95*

http://www.portal.com/~cyberlaw/cylw0195.html

[18] *Internet Threat Story Prompts Response*

http://www.mecklerweb.com/simba/95021601.htm

[19] *ALET: S314 Online "Decency Act" Threatens All Online Providers*

http://www.eff.org/pub/Alerts/eff_s314_hr1004.statement

[20] *Intellectual Property and the National Information Infrastructure*

http://www.uspto.gov/text/pto/nii/ipwg.html

[21] *Xanadu home page*

`http://www.aus.xanadu.com:70/1/xanadu`

[22] Howard Rheingold, *The Virtual Community: Homesteading on the Electronic Frontier*, HarperPerennial Paperback, 1993

`http://www.well.com/www/hlr/vcbook/index.html`

[23] *CommerceNet: The First Large-Scale Market Trial of Electronic Commerce on the Internet*

`http://www.commerce.net/information/background.html`

[24] *MarketNet Homepage*

`http://www.computek.net/mktnet/index.html`

[25] *MarketPlace.com*

`http://marketplace.com/`

[26] *Dealernet*

`http://www.dealernet.com/index.html`

[27] *Internet Business Directory*

`http://ibd.ar.com/`

[28] *Open Market Home Page*

`http://www.openmarket.com/`

[29] *Shopping2000 Home Page*

`http://www.shopping2000.com/`

[30] *NetMarket Homepage*

`http://netmarket.com/nm/pages/home`

[31] *CyberMalls*

`http://www.cybermalls.com/cymont/cymonmal.htm`

[32] *The Internet Mall*

`http://www.mecklerweb.com/imall/imall.htm`

[33] *GNN Business Pages*

`http://gnn.com/gnn/bus/index.html`

[34] *An Overview of InfoMall*

`http://www.npac.syr.edu/infomall/MallInfo.html`

[35] *NSF Awards Digital Libraries Research*

`http://www-diglib.stanford.edu/diglib/pub/nsf.announce.html`

[36] *Secure-HTTP Press Release*

`http://www.terisa.com/pr/jv.html`

[37] *Spyglass(R) Adds Six New OEM Customers as Mosaic Partners*

`http://www.spyglass.com/newcuspr.htm`

[38] *Secure NCSA Mosaic Manual*

`http://www.commerce.net/software/SMosaic/Docs/manual.html`

[39] *The SSL (Secure Sockets Layer) Protocol*

`http://home.netscape.com/info/SSL.html`

[40] *Mosaic In A Box Press Release*

`http://www.spry.com/sp_corp/sp_annc/pr_mbox.html`

[41] *WWW Security Reference*

`http://www-ns.rutgers.edu/www-security/reference.html#other_lists`

[42] *HTTP Security: The IETF HTTP Security Working Groupe Home Page*

`http://www-ns.rutgers.edu/www-security/https-wg.html`

[43] *PowerPC News: Burn those bank notes - Digital cash is coming*, July 22, 1994

`http://power.globalnews.com/articles/3032.htm`

[44] *First Virtual (TM) Home Page*

`http://www.fv.com/`

[45] *Buying Information with First Virtual*

`http://www.fv.com/user/buyerdetails.html#whocanplay`

[46] *InfoHaus By-Seller Page*

`http://www.infohaus.com/access/by-seller`

[47] *CMU & Visa press release*, Feburary 15, 1995

`http://www.ini.cmu.edu/netbill/pr.html`

[48] Marvin Sirbu, J.D. Tygar *NetBill: An Internet Commerce System Optimized for Network Delivered Services*

`http://www.ini.cmu.edu/netbill/CompCon.html`

[49] Computer Emergency Reaponse Team 1994 Anuual Report (Summary)

`http://www.sei.cmu.edu/SEI/programs/cert/1994_CERT_Summary.html`

[50] 今泉大輔, 「電子貨幣 *ecash* は日本で可能か」, 金融ビジネス、1995年3月

[51] Marvin Sirbu, J. D. Tygar, *NetBill: An Internet Commerce System Optimized for Network Delivered Services*

[52] L.Jean Camp, Marvin Sirbu, J. D. Tygar, *Privacy, Anonymity and Reliability in Electronic Currency Transactions*, January 1, 1995

[53] David M. Kristol, Steve H. Low, Nicholas F. Maxemchuck, *Anonymous Internet Mercantile Protocol*, March 17, 1994

[54] *About the Digital Notary System*

`http://www.surety.com/about-dns.html`

[55] David Chaum, *Achieving Electronic Privacy*, Scientific American, August 1992, p. 96-101

`http://www.digicash.com/publish/sciam.html`

[56] David Chaum, *Prepaid Smart Card Techniques*

`http://www.digicash.com/publish/cardcom.html`

[57] *Digicash - Numbers that are money*

`http://www.digicash.com/publish/digibro.html`

[58] Steven Levy, *E-Money (That's What I Want)*

`http://www.wired.com/Lib/Wired/2.12/features/emoney.html`

[59] Jason Solinsky, *An Introduction to Electronic Commerce*

[60] David Chaum, *Online Cash Checks*, Advances in Cryptography, EUROCRYPTO789, pp. 288-293

`http://www.digicash.com/publish/online.html`

[61] Adi Shamir, *How to Share a Secret*, Communications of the ACM, November 1979, Volume 22, Number 11

[62] Stefan Brands, *An Efficient Off-line Cash System Based On The Representation Problem*

[63] S. Brands, *Off-Line Transfer by Smart Cards*

[64] *LETSystems - the Home Page*

`http://www.u-net.com/gmlets/`

[65] David M. Kristol, Steven H. Low, Nicholas F. Maxemchuk, *Anonymous Internet Mercantile Protocol*

[66] Steven H. Low, Nicholas F. Maxemchuck, *Anonymous Credit Cards*, 2nd ACM Conference on Computer and Communication Securit, Nov. 2-4, 1994

[67] B. Clifford Neuman, Gennardy Medvinsky, *Requirement for Network Payment: The Netcheque Perspective*, Proceeding of IEEE Compcon'95, San Francisco, Mark 1995

[68] Gennardy Medvinsky, B. Clifford Neuman, *NetCash: A design for practical electronic currency on the Internet*

[69] B. Clifford Neuman, Gennardy Medvinsky, *Requirements for Network Payment: The NetCheque Perspective*, Proceedings of IEEE Compcon'95, San Francisco, March 1995

[70] *Welcome To NetChex*

`http://www.netchex.com/`

[71] *Electronic Currency for the Internet*, Electronic Markets, No. 9-10, October 23, Page 23-24

[72] B. Clifford Neuman, *Proxy-Based Authorization and Accounting*, Proceedings of the 13the International Conference od Distributed Computing Systems, May 1993

[73] *CyberCash Home Page*

`http://www.cybercash.com/`

[74] *XIWT Cross-Industry Working Team*

`http://www.cnri.reston.va.us:3000/XIWT/public.html`

World Wide WebのHelper Applicationインタフェース MIME TypeとMailcap

大久保清貴

1995年3月22日

1 MIME Type

Mosaicを使っていてページの内容がイメージだとビューアーとしてxvコマンドが立ち上がる。オーディオだとshowaudioコマンド立ち上がる。パッドを持ってきたときIntelligentPadが立ち上がるのも同じ仕組みを使っている。ファイルのタイプに応じてMosaicが呼び出すコマンドのことをHelper Applicationという。どのHelper Applicationを立ちあげるかを決定するのに関係があるファイルが2つあって、mime type、mailcapと呼ばれる。

MIMEとはMultipurpose Internet Mail Extensionsの略で、電子メールの内容にマルチメディア・ドキュメントを含められるようにするための拡張である。これはRFC1521に定められている[1]。RFCとはRequest For Commentの略でインターネットのプロトコルを定めた文書のシリーズを表す。MIMEは電子メールばかりでなく、World Wide Webで扱われるマルチメディア・ドキュメントのためにも使われる。WWWは、クライエント／サーバに分かれて動いている。したがって、例えばサーバがイメージを送る場合に、クライエント側のアプリケーションとしてxvを指定することはできない。なぜなら、クライエントはUNIXマシンとは限らず、Macintoshあるいは Windowsかもしれないからである。またクライエントを使っているユーザごとに立ちあげたいアプリケーションが異なるかもしれない。

そこでWWWではファイル・タイプとアプリケーションの対応テーブルを1個所で管理しないで2個所に分けている。このテーブルは、サーバ側ではMIME type、クライエント側ではmailcapと言われる。MailcapについてはRFC1524[2]で標準が定められている。

2 ファイル・タイプによるHelper Applicationの起動

図1にMosaicでパッドを指すURL(Universal Resource Locator)を持つアンカーがクリックされてからIntelligentPadが立ち上がるまでのプロセスを示す。

1. Mosaicのアンカーをクリックして`example.pad`をサーバに要求する。
2. サーバは`mime.types`というテーブルから、ファイル名の拡張子padをもつものを捜して、パッドのタイプが`application`でサブタイプが`x-iploader`であることをみつける。
3. サーバはファイルとともにタイプ／サブタイプ情報をクライエントに送る。このとき、ファイルは`/tmp`などの下にテンポラリとして作られる。

World Wide Web の Helper Application インタフェース
MIME Type と Mailcap

大久保清貴

1995年3月22日

1 MIME Type

Mosaic を使っていてページの内容がイメージだとビューアーとして xv コマンドが立ち上がる。オーディオだと showaudio コマンドが立ち上がる。パッドを持ってきたとき IntelligentPad が立ち上がるのも同じ仕組みを使っている。ファイルのタイプに応じて Mosaic が呼び出すコマンドのことを Helper Application という。どの Helper Application を立ち上げるかを決定するのに関係があるファイルが2つあって、mime.types、mailcap と呼ばれる。

MIME とは Multipurpose Internet Mail Extensions の略で、電子メールの内容にマルチメディア・ドキュメントを含められるようにするための規格である。これは RFC1521 に定められている[1]。RFC とは Request for Comment の略でインターネットのプロトコルを定めた文書のシリーズを表す。MIME は電子メールばかりでなく、World Wide Web で扱われるマルチメディア・ドキュメントのためにも使われる。WWW は、クライエント／サーバーに分かれて動いている。したがって、例えばサーバーがイメージを送る場合に、クライエント側のアプリケーションとして xv を指定することはできない。なぜなら、クライエントは UNIX マシンとは限らず、Macintosh あるいは Windows かもしれないからである。またクライエントを使っているユーザーごとに立ち上げたいアプリケーションが異なるかもしれない。

そこで WWW ではファイル・タイプとアプリケーションの対応テーブルを1箇所で管理しないで2箇所に分けている。このテーブルは、サーバー側では MIME type、クライエント側では mailcap と言われる。Mailcap については RFC1524[2] で標準が定められている。

2 ファイル・タイプによる Helper Application の起動

図1に Mosaic でパッドを指す URL(Universal Resource Locator) を持つアンカーがクリックされてから IntelligentPad が立ち上がるまでのプロセスを示す。

1. Mosaic のアンカーをクリックして `example.pad` をサーバーに要求する。
2. サーバーは `mime.types` というテーブルから、ファイル名の拡張子 `pad` をもつものを探して、パッドのタイプが `application` でサブタイプが `x-uploader` であることをみつける。
3. サーバーはファイルとともにタイプ／サブタイプ情報をクライエントに送る。このとき、ファイルは /tmp などのテンポラリとして作られる。

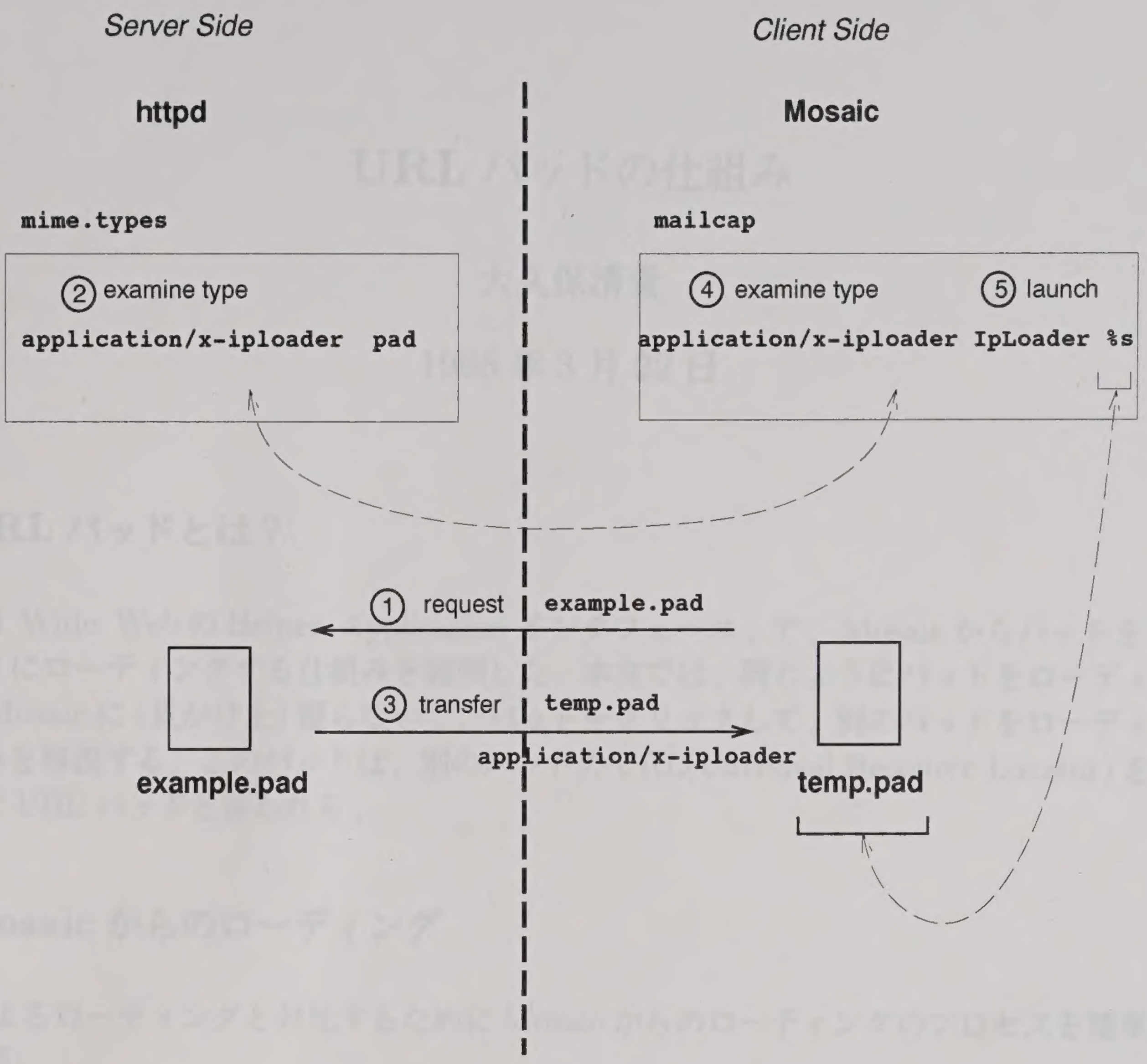

Server Side
Client Side
httpd
Mosaic
mime.types
② examine type
application/x-iploader pad
mailcap
④ examine type
⑤ launch
application/x-iploader IpLoader %s
① request
example.pad
③ transfer
temp.pad
application/x-iploader
example.pad
temp.pad

図 1: mime type と mailcap

4. クライエントは、`mailcap` テーブルを調べて、`application/x-iploader` に対してどのアプリケーションを立ちあげればよいか決める。
5. 決定したアプリケーション `IpLoader` をその引数にファイル名 `temp.pad` を付けて起動する。

参考文献

[1] N. Borenstein, N. Freed, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, RFC1521

[2] N. Borenstein, *A User Agent Configuration Mechanism For Multimedia Mail Format Information*, RFC1524

URL パッドの仕組み

大久保清貴

1995 年 3 月 22 日

1　URL パッドとは？

「World Wide Web の Helper Application インタフェース」で、Mosaic からパッドを IntelligentPad にローディングする仕組みを説明した。本文では、同じようにパッドをローディングするが、Mosaic に (見かけ上) 頼らないで、パッドをクリックして、別のパッドをローディングする仕組みを解説する。このパッドは、別のパッドの URL(Universal Resource Locator) を持っているので URL パッドと言われる。

2　Mosaic からのローディング

URL によるローディングと対比するために Mosaic からのローディングのプロセスを簡単に示す (図 1 参照)。

1. request

   Mosaic のアンカーをクリックすると、WWW サーバにパッドの転送を依頼する。

2. transfer

   WWW サーバはセーブ形式のパッドを Mosaic の動いているマシンに転送する。

3. load

   IntelligentPad のパッドをロードするルーチンが呼ばれる。

3　URL パッドからのローディング

URL パッドを使う場合も、Mosaic からのローディングとほとんど同じである。違いは、ステップ 0 で Line Mode WWW Client を立ち上げて、Mosaic の代わりをさせるだけである。

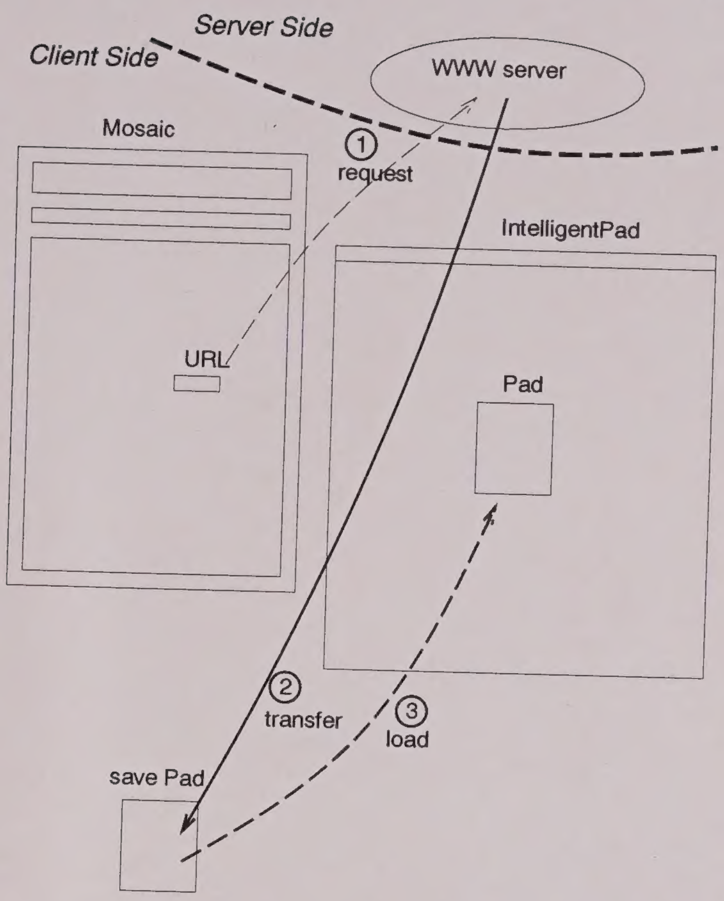

Server Side
Client Side
WWW server
Mosaic
① request
IntelligentPad
URL
Pad
② transfer
③ load
save Pad

図 1: Mosaic からのローディング

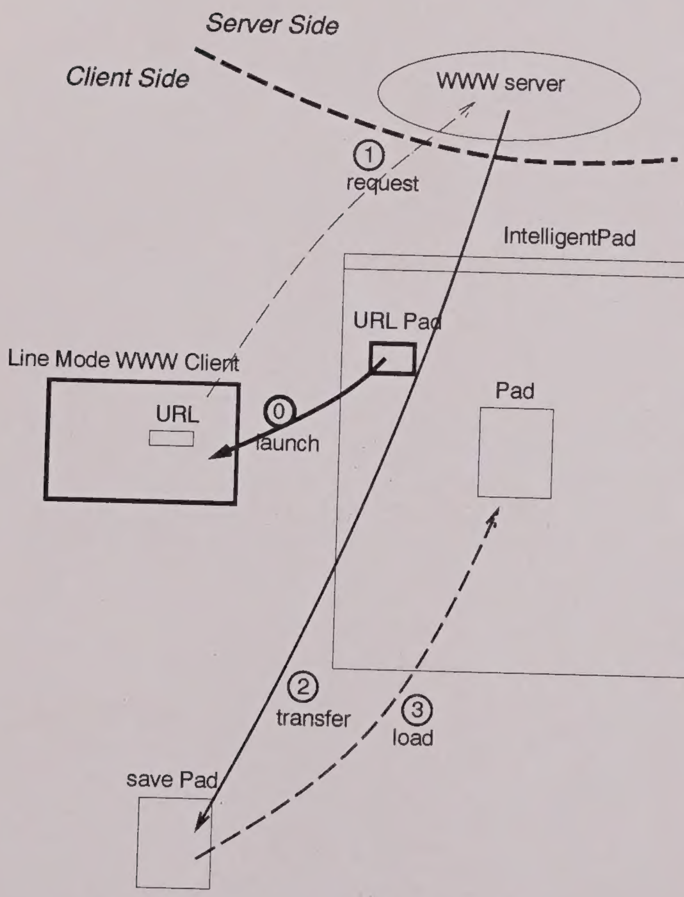

Server Side
Client Side
WWW server
① request
IntelligentPad
URL Pad
Line Mode WWW Client
URL
⓪ launch
Pad
② transfer
③ load
save Pad

図 2: URL パッドからの ローディング

XANADU®
REBORN
94.9.25

Sapporo
HyperLab

Welcome to the HyperLab.

OPEN
HOUSE
94.3.2[illegible]

Five papers on

ZIPS AND XANADU*

by Ted Nelson.

1. **Zip Editing, Xanadu Overview, Xanadu Publishing (1995)**
An overview of the general ideas of Zips and Xanadu.

2. **The Heart of Connection: Hypermedia United by Transclusion (1995).** (Near-final version, title may not be final.)
An invited paper that will appear in a special hypermedia issue of the *Communications of the Association for Computing Machinery* (CACM).

3. **Transcopyright: Pre-Permission for Virtual Republishing (1995)**
The copyright system of Project Xanadu, now presented as a separate legal concept.

4. **A File Structure for the Complex, the Changing, and the Indeterminate (1965)**
The original 1965 paper on zipper lists, hypertext and hypermedia.

5. **HyperLab Zip Editor: Preliminary Specifications (1995)**
The preliminary design of the "Tatsuno Otoshigo" (Sea Horse) zip editor we are working on.

Theodor Holm Nelson *Founder of Interactive Media* *Founder of Network Publishing*
Project Xanadu® On-Line Storage and Publishing Agency
3020 Bridgeway #295, Sausalito CA 94965 415/ 331-4422 Fax 415/ 332-0136
e-mail ted@xanadu.net
and
Sapporo HyperLab Doshin Maruyama, 8F • 2-20, Odori Nishi 20-chome
Chuo-Ku, Sapporo-shi, Hokkaido 064 JAPAN
Tel. (from the States) 011 81 11 621-1133 Fax (from the States) 011 81 11 621-1170
e-mail ted@xanadu.net

* "Xanadu" and the Eternal-Flaming-X Symbol are registered trademarks of Project Xanadu, Sausalito, California.

ZIP EDITING, XANADU* OVERVIEW, XANADU PUBLISHING

Ted Nelson, Research Fellow, Sapporo HyperLab

Preliminary Remarks.

THE GREAT HOPE OF COMPUTERS

The great hope of the computer age is for improved understanding, visualization and creative control of our work and our lives. THERE ARE MANY COMPLEX STRUCTURES AND IDEAS IN THE WORLD TO SEE AND UNDERSTAND. How can we see and understand them? How can we share them?

MEDIA MODELS OF SOFTWARE

Media are systems of construction that people create works in, works which have points of view. Writing, photography, architecture and cooking are media. But because they may be used more exactly to express complex ideas, writing and movies are deeper media.

Literature and movies are my basic models for software and computer media.

LITERATURE: literature is what's left of all the experience and research of a life, or a society. It consists of packages of information-- books, poetry, videotapes and many more types-- with points of view and connections

MOVIES: movies are events on a screen that affect the heart and mind of the viewer. Therefore software is A BRANCH OF MOVIE-MAKING, since software consist of events on a screen that affect the mind and heart of the viewer, PLUS INTERACTION.

THE GREAT HOPE OF ELECTRONIC PUBLISHING

The great hope of electronic publishing is that millions of people will have access to more ideas and knowledge than ever before, under conditions of freedom and accessibility and low price; that people will be able to tie things together, connecting information in whatever ways they wish for their own use, and re-publishing such connections for others.

But most people's experience with digital media has so far been an experience of difficulty, incompatibility and unconnectability; and most corporate plans and research designs of software still do not allow the kind of deep connectability that concerns me.

(continued)

*Xanadu®, and the Eternal Flaming X symbol®, are registered trademarks of Project Xanadu.

Trans©1994 Theodor Holm Nelson.

XANADU OVERVIEW

For me these hopes, and views of media, converge on one central idea--the system I call Xanadu (or sometimes by other names). This idea is at the center of all my thinking about media, ideas and electronic publishing.

I see Xanadu as AS THE BASIC INFORMATION STRUCTURE NEEDED IN TOMORROW'S WORLD.

It has been designed to have the greatest generality in directions which I think are important, serving up connected electronic documents in a way that is intended to be simple and democratic and clarifying.

A PARTICULAR FABRIC OF RELATIONS

Xanadu structure consists of a particular fabric of relations, like Lego(tm) building blocks, which can be combined together in specific ways.

The Parts: DOCUMENTS AND CONNECTIONS

There are two parts to the Xanadu fabric, documents and connections. But you can think of it another way: the connections are actually all stored inside the documents, so in the simplest view there are REALLY ONLY DOCUMENTS-- Xanadu connections are not separate parts, but reach out from inside the documents to touch other ones.

Anyone may create documents and connections between them. My connections, which are stored inside my documents, do not interfere with your connections, which are stored inside your documents.

A document may consist of any combination of data types (text, audio, pictures, movies, animations, 3D figures and so on); and any connections, either within the document or reaching out to other documents. The connections are of two types:

1. TRANSCLUSIONS. A transclusion means "bring that part in here." (Technically, it is a virtual inclusion or an embedded instance). It is not a copy, but an apparition of the original. I have discussed transclusion at length in many places (see paper, "The Heart of Connection").

2. LINKS. A link is any other relation that an author cares to point out between two things, whether it is a comment, an electronic sticky note or personal annotation, a disagreement, or an electronic staple that makes two things into one.

A PARTICULAR TYPE OF PUBLISHING

Xanadu publishing is electronic publishing that offers the ability to republish materials freely into new contexts, and to examine the same material side-by-side in these many different contexts.

Xanadu electronic documents may be purchased, viewed and kept by anyone. They may also be published by anyone: anyone may publish new material (bytes); anyone may publish connections to other people's documents (remember, the connections are inside documents that you publish). And anyone may REpublish virtually, by a particular method I now call Transcopyright (see attached paper, "Transcopyright: Pre-Permission for Virtual Republishing").

PUBLISHED LINKS.

A published link is some comment, or other relation, that the author creates or points out. A link, just like a paragraph of text, represents the point of view of the author, embodying the vision of the author who is making the link

PUBLISHED TRANSCLUSIONS.

A published transclusion is is a Xanadu connection that means "buy that and put it right here." This method allows everyone to include other publishers' materials virtually in their own documents, with payment and credit going to the original publisher.

EXPANDABLE AND ORDERLY

I believe this is an orderly, expandable and understandable system, a system that people of all ages and levels of education can learn to use easily, like books-- although of course some electronic documents may be difficult to read, like many books.

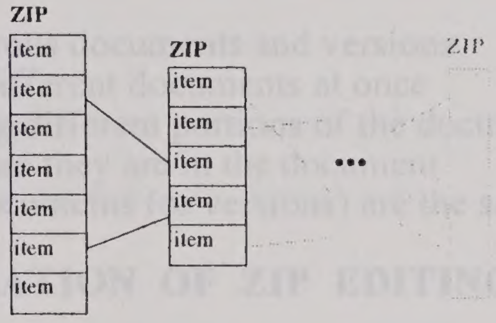
ZIP OVERVIEW

Zips are an abstract simplification of the Xanadu idea that focusses on central problems of visualization, understanding, manipulation and transclusion.

Thus I see zips as the simplest example of tomorrow's basic information structure.

WHAT ARE ZIPS?

Zips are connectable structures-- especially, structures whose parts may be cross-connected. In particular, zips are sequences of items which may be cross-connected betwen their parts..



WHAT IS ZIP EDITING?

Zip editing is working on such connected sequences of items, either reworking the pieces themselves or the overall structure.

WHAT ARE ZIPS FOR?

For keeping track of--
- overlapping lists
- documents and their outlines
- MULTIPLE USES OF AN ITEM
- versions of text
- versions of programs
- comments

and many other uses.

WHAT IS THE PURPOSE OF ZIP EDITING?

The purpose of zip editing is to see, work with and understand complex relations among--
- STRUCTURES
- PIECES OF STRUCTURES
- CROSS-INDEXING AND CROSSFILING
- RE-USES OF MATERIAL-- AND THEIR CONTEXTS
- ELECTRONIC PUBLISHING WITH RE-USE
- VARIANTS
- VERSIONS
- COMMENTS

Semi-final version.

Theodor Holm Nelson

THE HEART OF CONNECTION:

Hypermedia United by Transclusion

Invited paper to appear in
***Communications of the Association for Computing Machinery* (CACM),**
sometime in 1995,
opposite a similar paper by Douglas C. Engelbart.

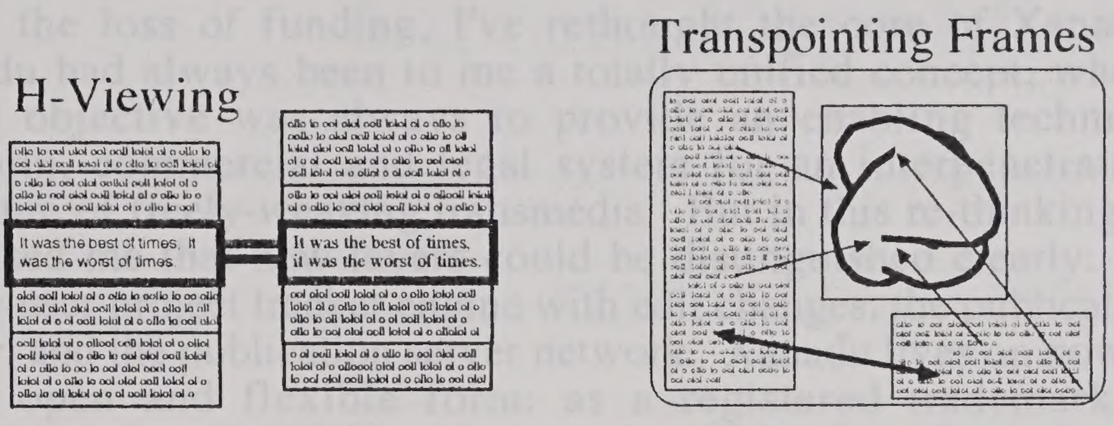
I can imagine few honors greater than a facing page to Douglas Engelbart's; he is one of the truly great men of our century and I love him dearly. Like Doug I have had a unified vision of which today's popular attainments are only centerless shadows of the edge.

Let me begin again, for few have understood.

I have till now had little basis for conversation with the computer mainstream. There has been a great expository problem, for which I have had to coin various terms (e.g., hypertext, hypermedia, transclusion) or import them. My plan after college was to be a rogue academic (like Fuller), writer and independent film-maker. My fundamental personal challenge was managing simultaneous projects in a massively parallel fashion. Then I faced, in graduate school, a computer course. As an abstract philosopher I recognized the computer first as a philosophy tool that dealt with arbitrary abstractions; as a moviemaker I recognized it next as a presentational device.

By the end of 1960 I had a design, and a paradigm. The central idea was what I now call *transclusion*. Simply stated, transclusion enables *re-use*, and more importantly, *intercomparing* what one reuses. Transclusion

keeps re-used items separate and identified, each with a pathway to its original. All instances of the same material are linked sideways, giving access to all uses of any item. Original context would always be recognizable. And I developed a method for selling quoted material that provides precise proportional royalty to the original publisher.


Transpointing Frames
H-Viewing
It was the best of times, it was the worst of times.
It was the best of times, it was the worst of times.

Two Ways of Viewing Transclusion

Now I needed a generalized server featuring intercomparison at the center of *all digital function*. The server would be *hyperarchical*, permitting the same material to be organized at the same time with simultaneous alternative structures---sequences, hierarchies, etc. This would provide writers and scholars with a personal revision server and project tracker, managing re-use with instance comparison in all projects and publications. This meant an author could try many possible organizational strategies. *No connections need be cut.* Thus work could unfold in its unified fullness. While the designed from the beginning as a single-user system, this also could be a system for workgroups; and indeed, could scale up directly to a mighty publishing system.

Thus XANADU* has always been an *integrated concept*, centering on *the parallel intercomparison of identified re-use*, and its uses both for personal work environments (managing private versions) and for hypermedia publishing---that is, all publishing---under matching copyright doctrine of sharable re-use. Each republished piece would be a tracked instance of the original. But the republisher would just make a transclusion pointer to each desired quotation. Users would pay royalties, based on what they received, to the original publisher. Xanadu has ALWAYS been a commercial project, intended as a licensed network of servers standing under independent local ownership.

Since 1960, half a dozen server designs have been implemented to

various degrees, progressively improving granularity, especially editability of embedded instances; mechanisms of transclusion; mechanisms of versioning; and transvisibility, or the ability to reach different instance portions from one another. Substantial funding was provided by Autodesk, Inc., from 1988 to 1992 and then dropped.

Since the loss of funding, I've rethought the core of Xanadu. Xanadu had always been to me a totally unified concept, whose grand objective was always to provide an enabling technical platform, commercial and legal system for an interpenetrating literature of freely-weaving transmedia. But in this re-thinking, it surprised me that new layers could be distinguished clearly: the single-user project tracker engine with edit changes, the publication server, and the publication server network. Xanadu lives on now in more open and flexible form: as a registered trademark, a publishing doctrine [1,2] and a set of specifications for software that will fit this publishing model [3,4] intended to guide the assembling of new component-based Xanadu software from pieces on the open market.

Like Doug Engelbart, I slog on with ideals held high above the mud. Telling is the irony that Doug and I started from completely opposite premises: he to empower work groups and make them smarter, I to free the individual from group obtuseness and impediment. That Doug and I have converged on common ideas to any degree is astonishing, and shows, I think, the correctness and the generality of the ideas.

* "Xanadu" is a registered trademark of Project Xanadu.

Bibliography
----

1. Nelson, T., "A File Structure for the Complex, the Changing and the Indeterminate." Proceedings of the ACM Twentieth National Conference, 1965.

2. Nelson, T. "Xanadu Publishing Doctrine," trans(c)1995 T. Nelson, listed Xanadu. Available also by anonymous ftp from [addr. before press time].

3. Nelson, T., "Publishing in the Point-and-Click Universe." Proceedings of the 1994 Australian Multimedia Conference [??]

4. Nelson, T., "The Xanadu Spec Book," trans(c)1995 T. Nelson, listed Xanadu. Available also by anonymous ftp from [addr. before press time].

Preliminary Short Version.

Transcopyright: Pre-Permission for Virtual Republishing

Trans© 1994 Theodor Holm Nelson. Xanadu®, and the Eternal-Flaming-X symbol®, are registered trademarks of Project Xanadu.

Theodor Holm Nelson *Founder of Interactive Media Founder of Network Publishing*
Project Xanadu® On-Line Storage and Publishing Agency
3020 Bridgeway #295, Sausalito CA 94965 415/ 331-4422 Fax 415/ 332-0136
e-mail ted@xanadu.net
and
Sapporo HyperLab Doshin Maruyama, 8F • 2-20, Odori Nishi 20-chome
Chuo-Ku, Sapporo-shi, Hokkaido 064 JAPAN
Tel. (from the States) 011 81 11 621-1133 Fax (from the States) 011 81 11 621-1170
e-mail ted@xanadu.net

SUMMARY

The on-line copyright problem may be resolvable by a simple, sweeping permission method. The proposed system, which anyone may use, allows broad re-use of materials in exchange for automatic tracking of ownership. Payment goes to the original publisher and credit to the original author. Nothing is misquoted and nothing is out of context-- since the original context is immediately available.

A wording and an abbreviation for this permission system are proposed here.

INTRODUCTION

Today's central controversy seems to be the question of how to manage copyright and royalty on the sale of digital content.

The standard question has been, "How do we prevent infringement?" If we re-frame the question as "How can we allow re-use?", the solution may be simpler and more powerful than everyone thinks, with benefits for everyone.

The idea described here occurred to the author in the fall of 1960, and since then has been an implicit part of the Xanadu Project (1), an ambitious system of hypermedia publishing servers with deep version management. Because the project has been controversial and beset with difficulties, our fundamental copyright proposal was not understood. And because this method was embedded in the software that had been designed for the Xanadu project, it was not recognized or expounded as a separate legal permission doctrine.

In this paper, we now separate that copyright method from its software history, and present it as a method anyone can use. It is a legal permission doctrine with far-reaching consequences, whose benefits may be considerable.

THE OBJECTIVE

From the beginning, the objective of Project Xanadu was to facilitate a new kind of literature: a new populist medium, a many-to-many publishing system, not centralized in editorial and publishing companies, but open to anyone. It was intended to create a new kind of freedom of the press, and make possible a deeper understanding of published materials (2).

But this would not mean simply electronic availability of closed documents; we would also need *re-use* of existing materials, which the computer could help track.

The idea was to preserve integrity, copyright and royalty for digital materials, yet allow everyone freely to re-use these materials, which would retain their identity at all times. By allowing everyone to republish all materials, this would effectively define a new system of fluid, digital transmedia, with universal freedom of re-use and visualization. All different republications could in principle be viewed side-by-side by the user, thus leading to greater understanding of the material and of the points of view of the different documents, their authors and republishers.

The scheme is simple: everyone is free to republish digital materials virtually, as quotations, anthologies and collages, provided that ***each copy of previously-published bytes is separately purchased from the original publisher at the time of delivery***. A republisher only distributes pointers showing how to obtain the material, and place it in the new contexts; each recipient buys these materials independently.

This method provides each receiving party with a properly-owned, independent copy. What may be just as important, it provides incentives for participation and a method for honest re-use in a system from which all benefit.

A PERMISSION DOCTRINE

This turns out actually to be a permission doctrine (just as "shareware" and "all rights reserved, except that churches may reprint freely") are permission doctrines.

In this permission doctrine, the copyright holder gives permission for republication, PROVIDED THAT: the material is sent to users only as an address for the material being republished, indicating what material is to be included in what new context; each recipient purchases the material anew from its respective publisher; the material is sent with identifiers as to its origin; and the user has software enabling these identifiers to be kept (so that, in principle, any recipient may republish in the same fashion, ad infinitum).

The copyright holder must therefore make a statement like the following:

"Permission is granted to all parties in the universe to re-publish the bytes of these materials virtually in new on-line contexts, provided that (1) such virtual republication consists of transmitting only an indication of what bytes are to be included in the particular new context, inviting the recipient to purchase the content bytes anew from the present publisher; and (2) that the present publisher is notified of any such virtual republication."

This may be adequate to achieve the desired result, as we will consider below. It also leaves room for a number of possible variations.

IRONICALLY SIMPLE

This may seem over-simple, when mighty technical and legislative solutions are being sought by so many parties.

But the real issue is context relinquishment -- the publisher's willingness to allow the material to be used in unpredictable contexts. This is a tradeoff. We expect publishers, especially small, non-traditional publishers and self-publishing individuals-- to be invigorated by the notion of their materials being available for re-use.

IMPLIED ASPECTS

Implied in this method are certain other aspects. For instance, the user must get a receipt for each portion purchased, so that the received materials may be proved a legitimate copy. A survivable memorandum of origin must accompany the repurchased materials, making virtual republication possible once again (by the new recipient). And the republisher must notify the original publisher of the republication, so that these re-uses may be seen.

SUPPORTING SOFTWARE ASPECTS

This permission doctrine has certain minimal requirements for supporting software. Certain unusual software features are required:

Server-side software: the ability to sell materials in small amounts, ideally at proportional rates, delivering the materials with a receipt and identity notification.

Client-side software: 1. A client viewer is required, for both recipients and republishers, which may be used to purchase the bytes, and which retains the identity of bytes purchased. 2. Any

republisher must in addition have an editing program which retains the original addresses of the edited materials, even as they become cut down. The republisher's editing program must also prepare the document for skeleton transmission, stripping out the actual bytes and substituting the original addresses in the necessary format for their repurchase.

OWNERSHIP OF EACH FINAL COPY

The recipient who purchases the suggested materials owns a final copy, like any book, each portion legally purchased. Even if the materials are later withdrawn from sale, each copy already acquired in this way retains its ownership status as the possession of the recipient.

INTEROPERABILITY

Note that any party may offer transcopyright; so that different variations of this arrangement (see below), with different details of availability, may be mixed and matched in a given document. Regardless of such variations, a legitimate mixed copy is still owned by the final recipient.

THE PROPOSED FORMAT AND ITS LEGAL STATUS

I propound for this permission system an extremely simple presentational format. A publisher signifies participation by the following notice, in the place of the usual copyright notice:

Trans©1995 Irving Snerd

This is a shorthand form of the long-form permission stated earlier, so that the publisher is spared having to include it. (Some distribution agency or availability scheme should also be specified by the transcopyright grantor.)

LEGAL STATUS OF THIS FORMAT

It will be noted that the traditional copyright notice

©1995 Irving Snerd

is FULLY CONTAINED within the transcopyright notice, and so is still present.

However, note also that the traditional copyright notice no longer has its former force in law, since you now establish copyright automatically by the creation of a work; so the copyright notice is now regarded as being a courtesy and a convenience. Thus we are offering the transcopyright permission as a further courtesy, so that all parties will know how they may republish.

By what authority is this a shorthand? I hereby *declare* it to be such a shorthand form, just as "shareware" and "copyleft" symbolize the permission systems proposed by Bob Wallace and Richard Stallman. If others use the transcopyright notification with the intent given here, it will be recognized in law as being synonymous with the longer permission notice.

VARIATIONS NOT COVERED; DIFFERENT JURISDICTIONS

This arrangement is for on-line screen viewing and keepage of digital data, not for paper or other solid media.

A number of issues are not handled by this arrangement. These include byte modification (such as use of audio samples, morphing, bytewise graphical operations). It does not involve printout. It does not cover site licensing for multiple users. It does not include model releases, talent releases, or other forms of permission not strictly part of copyright. It does not state how long materials will be offered for sale.

Who settles these details? This is within the scope of variation of the idea: different forms of transcopyright may be offered by different parties-- individuals, companies, or other administrations. Any party offering transcopyright may bundle different kinds of permission into their own systems. (For instance, the contracts of Project Xanadu will have specific clauses with respect to length of availability, paper printout and site licensing.)

ENFORCEMENT AND BENEFITS

This system is not automatically enforced, but neither is any other feasible electronic copyright system. It is no more or less enforceable than any other. The fact that it defines a convenient method for honest behavior, satisfying principal objectives of both publisher and republisher-- no longer "infringer," but honest participant-- is a strong argument in its favor.

BIBLIOGRAPHY

1. Nelson, Theodor Holm, *Literary Machines*. Availabile from Mindful Press, 3020 Bridgeway #295, Sausalito CA 94965, USA.

2. Nelson, Theodor Holm, "The Art of Connection: Literature Unified by Transclusion." *Communications of the ACM*, forthcoming.

A File Structure for The Complex, The Changing and the Indeterminate

Theodor H. Nelson
Vassar College, Poughkeepsie, N.Y. 12601

Reprinted from
Association for Computing Machinery Proceedings of the 20th National Conference
with permission of ACM and the editor-publisher, Lewis Winner.
Minor corrections by the author.

© 1965 By The
Association for Computing Machinery

SUMMARY

THE KINDS OF FILE structures required if we are to use the computer for personal files and as an adjunct to creativity are wholly different in character from those customary in business and scientific data processing. They need to provide the capacity for intricate and idiosyncratic arrangements, total modifiability, undecided alternatives, and thorough internal documentation.

The original idea was to make a file for writers and scientists, much like the personal side of Bush's Memex, that would do the things such people need with the richness they would want. But there are so many possible specific functions that the mind reels. These uses and considerations become so complex that the only answer is a simple and generalized building-block structure, user-oriented and wholly general-purpose.

The resulting file structure is explained and examples of its use are given. It bears generic similarities to list-processing systems but is slower and bigger. It employs zippered lists plus certain facilities for modification and spin-off of variations. This is technically accomplished by index manipulation and text patching, but to the user it acts like a multifarious, polymorphic, many-dimensional, infinite blackboard.

The ramifications of this approach extend well beyond its original concerns, into such places as information retrieval and library science, motion pictures and the programming craft; for it is almost everywhere necessary to deal with deep structural changes in the arrangements of ideas and things.

I want to explain how some ideas developed and what they are. The original problem was to specify a computer system for personal information retrieval and documentation, able to do some rather complicated things in clear and simple ways. The investigation gathered generality, however, and has eventuated in a number of ideas. These are an information structure, a file structure, and a file language, each progressively more complicated. The information structure I call zippered lists; the file structure is the ELF, or Evolutionary List File; and the file language (proposed) is called PRIDE.

In this paper I will explain the original problem. Then I will explain why the problem is not simple, and why the solution (a file structure) must yet be very simple. The file structure suggested here is the Evolutionary List File, to be built of zippered lists. A number of uses will be suggested for such a file, in order to show the breadth of its potential usefulness. Finally, I want to explain the philosophical implications of this approach for information retrieval and data structure in a changing world.

I began this work in 1960, with no help from anybody. Its purpose was to create techniques for handling personal file systems and manuscripts in progress. These two purposes are closely related and not sharply distinct. Many writers and research professionals have files or collections of notes which are tied to manuscripts in progress. Indeed, often personal files shade into manuscripts, and the assembly of textual notes becomes the writing of text without a sharp break.

I knew from my own experiment what can be done for these purposes with card file, notebook, index tabs, edge-punching, file folders, scissors and paste, graphic boards, index-strip frames, Xerox machine and the roll-top desk. My intent was not merely to computerize these tasks but to think out (and eventually program) the dream file: the file system that would have every feature a novelist or absent-minded professor could want, holding everything he wanted in just the complicated way he wanted it held, and handling notes and manuscripts in as subtle and complex ways as he wanted them handled.

Only a few obstacles impede our using computer-based systems for these purposes. These have been high cost, little sense of need, and uncertainty about system design.

The costs are now down considerably. A small computer with mass memory and video-type display now costs $37,000; amortized over time this would cost less than a secretary, and several people could use it around the clock. A larger installation servicing an editorial office or a newspaper morgue, or a dozen scientists or scholars, could cost proportionately less and give more time to each user.

The second obstacle, sense of need, is a matter of fashion. Despite changing economies, it is fashionably believed that computers are possessed only by huge organizations to be used only for vast corporate tasks or intricate scientific calculations. As long as people think that, machines will be brutes and not friends, bureaucrats and not helpmeets. But since (as I will indicate) computers could do the dirty work of personal file and text handling, and do it with richness and subtlety beyond anything we know, there ought to be a sense of need. Unfortunately, there are no ascertainable statistics on the amount of time we waste fussing among papers and mislaying things. Surely half the time spent in writing is spent physically rearranging words and paper and trying to find things already written; if 95% of this time could be saved, it would only take half as long to write something.

The third obstacle, design, is the only substantive one, the one to which this paper speaks.

Let me speak first of the automatic personal filing system. This idea is by no means new. To go back only as far as 1945, Vannevar Bush, in his famous article "As We May Think"[1], described a system of this type. Bush's paper is better remembered for its predictions in the field of information retrieval, as he foresaw the spread and power of automatic document handling and the many new indexing techniques it would necessitate. But note his predictions for personal filing:

> Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is

> mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.
>
> It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk.
>
> ....
>
> A special button transfers him immediately to the first page of the index. Any given book of his library [and presumably other textual material, such as notes] can thus be called up and consulted with far greater facility than if it were taken from a shelf. As he has several projection positions, he can leave one item in position while he calls up another. He can add marginal notes and comments, ..." (1, 106-7)

Understanding that such a machine required new kinds of filing arrangements, Bush stressed his file's ability to store related materials in associative trails, lists or chains of documents joined together.

> When the user is building a trail, he names it, inserts the name in his code book, and taps it out on his keyboard. Before him are the two items to be joined, projected onto adjacent viewing positions. At the bottom of each there are a number of blank code spaces, and a pointer is set to indicate one of these on each item. The user taps a single key, and the items are permanently joined.
>
> ....
>
> Thereafter, at any time, when one of these items is in view, the other can be instantly recalled merely by tapping a button below the corresponding code space. Moreover, when numerous items have been thus joined together to form a trail, they can be reviewed in turn, rapidly or slowly, by deflecting a lever like that used for turning the pages of a book. It is exactly as though the physical items had been gathered together from widely separated sources and bound together to form a new book. It is more than this, for any item can be joined into numerous trails.
>
> ....
>
> Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. (1, 107)

Two decades later, this machine is still unavailable*.

The hardware is ready. Standard computers can handle huge bodies of written information, storing them on magnetic recording media and displaying their contents on CRT consoles, which far outshine desktop projectors. But no programs, no file software are standing ready to do the intricate filing job (keeping track of associative trails and other structures) that the active scientist or thinker wants and needs. While Wallace[3] reports that the System Development Corporation has found it worthwhile to give its employees certain limited computer facilities for their own filing systems, this is a bare beginning.

Let us consider the other desideratum, manuscript handling. The remarks that follow are intended to apply to all forms of writing, including fiction, philosophy, sermons, news and technical writing.

---

* The Bush Rapid Selector, which he designed (2), is a powerful microfilm instrument, but it is not suited to idiosyncratic personal uses, nor to evolutionary modification, as described hereunder.

The problems of writing are little understood, even by writers. Systems analysis in this area is scanty; as elsewhere, the best doers may not understand what they do. Although there is considerable anecdote and lore about the different physical manuscript and file techniques of different authors, literary tradition demerits any concern with technical systems as detracting from "creativity." (Conversely, technical people do not always appreciate the difficulty of organizing text, since in technical writing much of the organization and phraseology is given, or appears to be.) But in the computer sciences we are profoundly aware of the importance of systems details, and of the variety of consequences for both quality and quantity of work that result from different systems. Yet to design and evaluate systems for writing, we need to know what the process of writing is.

There are three false or inadequate theories of how writing is properly done. The first is that writing is a matter of inspiration. While inspiration is useful, it is rarely enough in itself. "Writing is 10% inspiration, 90% perspiration," is a common saying. But this leads us to the second false theory, that "writing consists of applying the seat of the pants to the seat of the chair." Insofar as sitting facilitates work, this view seems reasonable, but it also suggests that what is done while sitting is a matter of comparative indifference; probably not.

The third false theory is that all you really need is a good outline, created on prior consideration, and that if the outline is correctly followed the required text will be produced. For most good writers this theory is quite wrong. Rarely does the original outline predict well what headings and sequence will create the effects desired: the balance of emphasis, sequence of interrelating points, texture of insight, rhythm, etc. We may better call the outlining process inductive: certain interrelations appear to the author in the material itself, some at the outset and some as he works. He can only decide which to emphasize, which to use as unifying ideas and principles, and which to slight or delete, by trying. Outlines in general are spurious, made up after the fact by examining the segmentation of a finished work. If a finished work clearly follows an outline, that outline probably has been hammered out of many inspirations, comparisons and tests**.

Between the inspirations, then, and during the sitting, the task of writing is one of rearrangement and reprocessing, and the real outline develops slowly. The original crude or fragmentary texts created at the outset generally undergo many revision processes before they are finished. Intellectually they are pondered, juxtaposed, compared, adapted, transposed, and judged; mechanically they are copied, overwritten with revision markings, rearranged and copied again. This cycle may be repeated many times. The whole grows by trial and error in the processes of arrangement, comparison and retrenchment. By examining and mentally noting many different versions, some whole but most fragmentary, the intertwining and organizing of the final written work gradually takes place***.

Certain things have been done in the area of computer manuscript handling. IBM recently announced its "Administrative Terminal System"[5,6,7,8] which permits the storage of unfinished sections of text in computer memory, permits various modifications by the user, and types up the final draft with page numbers, right justification and headers.

While this is a good thing, its function for manuscripts is cosmetic rather than organizing. Such a system can be used only with textual sections which are already well organized, the visible part of the iceberg. The major and strenuous part of such writing must already have been done.

** I understand that this account is reasonably correct for such writers as Tolstoy, Winston Churchill and Katherine Anne Porter. Those who can stick to a prior outline faithfully, like James Fenimore Cooper, tend to be either hacks or prodigies, and don't need this system.

*** For a poignant, mordant portrayal of the writer's struggle, the reader is directed to Gorey's The Unstrung Harp; or, Mr Earbrass Writes a Novel.

If a writer is really to be helped by an automated system, it ought to do more than retype and transpose: it should stand by him during the early periods of muddled confusion, when his ideas are scraps, fragments, phrases, and contradictory overall designs. And it must help him through to the final draft with every feasible mechanical aid-- making the fragments easy to find, and making easier the tentative sequencing and juxtaposing and comparing.

It was for these two purposes, taken together-- personal filing and manuscript assembly-- that the following specifications were drawn up.

Here were the preliminary specifications of the system: It would provide an up-to-date index of its own contents (supplanting the "code book" suggested by Bush). It would accept large and growing bodies of text and commentary, listed in such complex forms as the user might stipulate. No hierarchical file relations were to be built in; the system would hold any shape imposed on it. It would file texts in any form and arrangement desired-- combining, at will, the functions of the card file, loose-leaf notebook, and so on. It would file under an unlimited number of categories. It would provide for filing in Bush trails. Besides the file entries themselves, it would hold commentaries and explanations connected with them. These annotations would help the writer or scholar keep track of his previous ideas, reactions and plans, often confusingly forgotten.

In addition to these static facilities, the system would have various provisions for change. The user must be able to change both the contents of his file and the way they are arranged. Facilities would be available for the revising and rewording of text. Moreover, changes in the arrangements of the file's component parts should be possible, including changes in sequence, labelling, indexing and comments.

It was also intended that the system would allow index manipulations which we may call dynamic outlining (or dynamic indexing). Dynamic outlining uses the change in one text sequence to guide an automatic change in another text sequence. That is, changing an outline (or an index) changes the sequence of the main text which is linked with it. This would permit a writer to create new drafts with a relatively small amount of effort, not counting rewordings.

However, because it is necessary to examine changes and new arrangements before deciding to use or keep them, the system must not commit the user to a new version until he is ready. Indeed, the system would have to provide spin-off facilities, allowing a draft of a work to be preserved while its successor was created. Consequently the system must be able to hold several-- in fact, many-- different versions of the same sets of materials. Moreover, these alternate versions would remain indexed to one another, so that however he might have changed their sequences, the user could compare their equivalent parts.

Three particular features, then, would be specially adapted to useful change. The system would be able to sustain changes in the bulk and block arrangements of its contents. It would permit dynamic outlining. And it would permit the spin-off of many different drafts, either successors or variants, all to remain within the file for comparison or use as long as needed. These features we may call evolutionary.

The last specification, of course, one that emerged from all the others, was that it should not be complicated.

These were the original desiderata. It was not expected at first that a system for this purpose would have wider scope of application; these jobs seemed to be quite enough. As work continued, however, the structure began to look more simple, powerful and general, and a variety of new possible uses appeared. It became apparent that the system might be suited to many unplanned applications involving multiple categories, text summaries or other parallel documents, complex data structures requiring human attention, and files whose relations would be in continuing change.

Note that in the discussion that follows we will pretend we can simply see into the machine, and not worry for the present about how we can actually see, understand and manipulate these files. These are problems of housekeeping, I/O and display, for which many solutions are possible.

Elements of the ELF

What was required we may call an evolutionary file structure: a file structure that can be shaped into various forms, changed from one arrangement to another in accordance with the user's changing need. It was apparent also that some type of list structure was necessary. Making the file out of lists would allow different categories of personal notes, separate drafts, outlines and master indices all to be handled as lists of some sort; their segments could then be manipulated through automatic handling of index numbers. The resulting file structure I will accordingly call the Evolutionary List File, or ELF, since it is an evolutionary file structure constructed with lists. The system proposed here is not the only ELF possible. It is built upon a specific technique of attaching lists together which has a natural resistance to becoming confused and messy.

As computer-based systems grow in capability and diversity of uses, they tend to become more and more cluttered with niggling complications, hidden passageways, and lurking, detailed interlocks, restrictions, specializations, provisos. These should be forsworn, if possible, in the system under discussion, so that it might be attractive to laymen (including artists and writers) who feel unkindly disposed toward computers. It should readily adapt to their own styles of handling things, imposing few conventions or methods of use. How could this imposition be avoided? And among so many interesting and possible system functions and file relations, how may the users know what connections to make, how may they understand what they are doing, and how may they avoid muddling and losing the things they are working with?

The answer, I think you see, is to choose a very simple structure that can be used and compounded in many different ways. The basic arrangement chosen for these purposes is an information structure I will refer to as zippered lists. (We might call it permutation-invariant one-for-one inter-list entry-linking, but that is not necessary.)

There are only three kinds of things in the zippered-list ELF, with no predetermined relations among them-- no hierarchies, machine-based features or trick exceptions. The system is user-oriented and open-faced, and its clear and simple rules may be adapted to all purposes.

The ELF has three elements: entries, lists and links. An entry is a discrete unit of information designated by the user. It can be a piece of text (long or short), a string of symbols, a picture or a control designation for physical objects or operations.

A list is an ordered set of entries designated by the user. A given entry may be in any number of lists.

A link is a connector, designated by the user, between two particular entries which are in different lists (Figure 1). An entry in one list may be linked to only one entry in another list.

On the left we see two zippered lists. Between the entries of list A and those of B are dashed lines, representing the links between the two lists. On the right is the table of links as it might look to a machine. The machine can read this table from right to left or left to right, finding entries in B that correspond to given entries in A, or vice versa. A change in the sequence of either list, or additions to either list, will not change the links that stand between them. Changes in the link structure will occur only if the user specifically changes the links, or if he destroys entries which are linked to others.

To be technical, then, two lists are zippered if there are any pairwise links between their respective elements, each element is in no more than one link pair, and these links are unaffected by permutation of the lists, remaining affixed to the same pairs of elements. It is not required that the two lists be of the same length, or, even if they are, that all entries have a link to the other list.

The ELF's File Operations

Zippered lists are an information structure; the Evolutionary List File is a file structure. The ELF described in this paper holds its contents exclusively as zippered (or unzippered) lists. But the file structure must also include a set of operations by which it may be modified. These file operations exist for creating, adjusting or removing the entry, list and link, and for manipulating the sequence relation. An ELF is actually any machine which will, on command, carry out the basic operations on entry, list, link and sequence.

Entries. The user may create new entries at any time, putting anything in them that he thinks appropriate. Entries may be combined or divided (unless indivisible, like objects, commands, etc.) Entries may be put in any list, and the same entry may be put in different lists. The user may direct that entries of one list be automatically copied onto another list, without affecting the original list.

Lists. The user may create lists and assign entries to them. He may at will make new copies of lists. He may rearrange the sequence of a list, or copy the list and change the sequence of that copy. Lists may be combined; lists may be cut into sublists.

Links. The user may create links between entries that are in different lists. Any number of legal links may be created, although the upper limit of links between any two lists is determined by the 1-for-1 rule. When an entry or a list is copied into a list, links will remain between parent and daughter entries. Moreover, after a list-copying operation, the daughter list will have the same links to all other lists as does the parent list.

Sequences. The user may put a list in any sequence he wishes. (A copied list will maintain the original sequence until modified.) Sequences may be transferred between lists via the links: if the sequence of A is transferred to B, each entry of A linked to an entry in B takes the sequential position of its linked

entry in B.

No definite meaning is assigned to these entities or operations by the system; the user is free to let them mean anything he likes. A list may be a category, trail, index, dialogue, catalogue or poem, and lists may be assembled into larger structures. The ELF may be thought of as a *place*; not a machine, but a piece of stationery or office equipment with many little locations which may be rearranged with regard to one another****.

Note that zippered lists generate only one of various possible Evolutionary List Files. Indeed, the description of the file structure given here is in some ways restrictive: the ELF could take a number of other, closely similar forms and still be much the same thing. For example, it would be possible to allow subentries and superentries into the file, to behave and link up like normal entries, even though they contained or were contained in other entries. But the equivalent can be done with the current system. Another possibility would be to allow links other than 1-for-1; these could be modal, the different link-modes having different meanings to the user. Or we might make it an evolutionary *network* file, allowing any two entries to be connected. Or, besides such general changes in the rules, plausible changes and accessory functions for any purposes could be introduced outside the given file structure, even including modifications and widgets to do some of the same things "more easily."

But to the user such complication might render the system far less handy or perspicuous. The ELF, with its associated techniques as described above, is simple and unified. Many tasks can be handled within the file structure. This means it can be of particular benefit to people who want to learn without complications and use it in ways they understand. For psychological, rather than technical reasons, the system should be lucid and simple. I believe that this ELF best meets these requirements.

Technical Aspects

Since the ELF description above bears some resemblance to the list languages, such as IPL, SLIP, etc., a distinction should be drawn. These list languages[9] are particularly suited to *processing* data, fast and iteratively, whose elements are manipulable in Newell-Shaw-Simon lists. Essentially they may be thought of as organizations of memory which facilitate sequential operations on unpredictably branching or hierarchical data. These data may change far too quickly for human intervention. Evolutionary file structures, and the ELF in particular, are designed to be changed piecemeal by a human individual. While it might be convenient to program an ELF in one of these languages, the low speed at which user file commands need to be executed makes such high-powered implementation unnecessary; the main problem is to keep track of the file's arrangements, not to perform computation on its contents. Although work has been done to accommodate the list-language approach to larger chunks of material than usual[10], the things people will want to put into an ELF will typically be too big for core memory.

The ELF does in fact share some of the problems of the list languages: not available-storage accounting or garbage collection (concerns associated with organization of fast memory for processing, which may be avoided at slower speeds), but the problems of checkout for disposal (what other lists is an entry on?) and list naming. The former problem is rather straightforwardly solved[11, p. 164]; the latter is complicated in ways we cannot go into here.

**** An ELF might even be constructed out of cards, blocks, sticks and strings, using techniques of puppetry, but this would not be a convenient object.

The ELF appears to be closest, topologically and in other organizing features, to the Multilist system described by Prywes and Gray[12]. Like that system, it permits putting entries in many different lists at once. However, in current intent[13] that system is firmly hierarchical, and thus somewhat removed from the ELF's scope of application. Another closely related system is the Integrated Data Store of Bachman[14,15,16,17,18]; this is intended as a hardware-software system for disc I/O and storage arrangement, but in its details it seems the ELF's close relative. Each of these systems has a connection logic that might be feasible as a basis for an ELF different from this one. Or, either might prove a convenient programming base for the implementation of this file structure.

Another obvious technical question must be considered. How can the ELF allow "unlimited" copies of entries and lists? By patching techniques, of course. Variant entries and lists can take virtually no space, being modification data plus pointers to the original. When a modified version of a list or entry is created, the machine patches the original with the changes necessary to make the modified version. (Figure 2)

USES

In the discussion that follows, we will examine various possible applications of zippered lists and the ELF, and postpone discussing the file language they require. Finally we will return to this problem, and describe the file language PRIDE whose additional features are needed to adapt the ELF for the uses originally discussed.

By assigning entries to lists, the ELF may be used as a glorified card file, with separate lists used for categories, trails, etc. This permits extensive cross-indexing by the assignment of one entry to different lists. It permits subsets and sub-sequences for any use to be held apart and examined without disturbing the lists from which they have been drawn, by copying them onto other, new lists. The ELF permits the filing of historical trails or associative (Bush) trails through documents, business correspondence, belles-lettres, case law, treaties, scholarly fields and history, and the mixture of trail with categorical filing.

These are the simple uses; the compound uses are much more interesting. But since we cannot intuitively fit every possible conceptual relationship into zippered lists, imaginative use is necessary. Remember that there is no correct way to use the system. Given its structure, the user may figure out any method useful to him. A number of different arrangements can be constructed in the ELF, using only the basic elements of entry, list and link. Zippered lists may be assembled into rectangular arrays, lattices and more intricate configurations. These assemblies of lists may be assigned meaning in combination by the user, and the system will permit them to be stored, displayed, taken apart for examination, and corrected, updated, or modified.

By using such combining arrangements on lists composed of text, the file can be self-documenting, with all labelling and documentation kept integrally within the file structure. It is thus possible to incorporate, in a body of information filed in the ELF, various levels of index, summary, explanation and commentary. Many useful ways of listing and linking such documentation are possible. In Figure 3 we see some of the ways that documentary lists may be linked together. The lists shown are outline, suboutline, draft, subdraft, summary, commentary and source list. These are not all the possible types of documentary lists; for example, "footnotes" are omitted. The ELF will permit any number of these documentary lists;

observe that they can be built on one another, and indefinitely compounded. The system will have no trouble accepting a commentary on a commentary on a subdraft of an outline for a variant list of source materials.

Figure 3 shows also how two lists may contain some of the same entries. The dashed line represents linkage between entries, the solid line shows that both lists contain the same entry. This may be useful for creating alternate versions, or, as in this example, the lists containing the same entry may have different purposes. Here, for instance, an entry in the summary is also to be found in the main draft.

This self-documentation feature permits any string of text in the ELF, long or short, to be annotated or footnoted for scholarly or other purposes. Such marginalia can be temporary or permanent, for the private memoranda of an individual or for communication among different persons using the file.

In a like manner, the ELF is capable of storing many texts in parallel, if they are equivalent or linked in some way. For example, instruction manuals for different models of the same machine may be kept in the file as linked lists, and referred to when machines are to be compared, used or fixed. This is of special use to repairmen, project managers and technical writers.

Moreover, the ELF's cross-sequencing feature -- the fact that links ignore permutations-- permits the collation of very different cognate textual materials for comparison and understanding. In law, this would help in comparing statutes (or whole legal systems); in literature, variorum editions and parodies. Thus such bodies as the Interpreter's Bible and a Total Shakespeare (incorporating Folios, bowdlerizations, satires and all critical commentary) could be assembled for study.

Let me try to illustrate the possible comprehensiveness and versatility of this file structure as applied to texts. Figure 4 shows the different arrangements that might be used by one man-- in this case an historian writing a book-- to assemble and integrate his intellectual and professional concerns. Although it is impossible to show the links between all the separate entries of these lists-- the entries are not themselves discernible in this drawing-- it is possible to note the kinds of links between lists. A thin line between lists shows that some links exist; a solid line indicates that some entries of both lists are the same.

Perhaps this looks complicated. In fact, each of the connectors shows an indexing of one body of information to another; this user may query his file in any direction along these links, and look up the parts of one list which are related to parts of another. Therefore the lines mean knowledge and order. Note that in such uses it is the man's job to draw the connections, not the machine's. The machine is a repository and not a judge.

The ELF may be an aid to the mind in creative tasks, allowing the user to compare arrangements and alternatives with some prior ideal. This is helpful in planning nonlinear assemblages (museum exhibits, casting for a play,) or linear constructions of any kind. Such linear constructions include not only written texts; they can be any complicated sequences of things, such as motion pictures (in the editing stage) and computer programs.

Indeed, computer programming with an on-line display and the ELF would have a number of advantages. Instructions might be interleaved indefinitely without

resorting to tiny writing. Moreover, the programmer could keep up work on several variant approaches and versions at the same time, and easily document their overall features, their relations to one another and their corresponding parts. Adding a load-and-go compiler would create a self-documenting programming scratchpad.

The natural shape of information, too, may call for the ELF. For instance, sections of information often arrange themselves naturally in a lattice structure, whose strands need to be separately examined, pondered or tested. Such lattices include PERT networks, programmed instruction sequences, history books and genealogical records. (The ELF can handle genealogical source documentation and its original text as well.) Indeed, any informational networks that require storage, handling and consideration will fit the ELF; a feature that could have applications in plant layout, social psychology, contingency planning, circuit design and itineraries.

The ELF may, through its mutability, its expansibility, and its self-documentation features, aid in the integration, understanding and channeling of ideas and problems that will not yield to ordinary analysis or customary reductions; for instance, the contingencies of planning, which are only partially Boolean. Often the reason for a so-called Grand Strategy in a setting is that we cannot keep track of the interrelations of particular contingencies. The ELF could help us understand the interrelations of possibilities, consequences, and strategic options. In a logically similar case, evaluating espionage, it might help trace consistencies and contradictions among reports from different spies.

The use of an ELF as the basis for a management information system is not inconceivable. Its evolutionary capability would provide a smooth transition from the prior systems, phasing out old paperwork forms and information channels piecemeal. Beginning with conventional accounting arrays and information flow, and moving through discrete evolutionary steps, the ELF might help restructure an entire corporate system. Numerical subroutining could permit the system to encompass all bookkeeping. The addresses of all transaction papers, zippered to lists of their dates and contents, would aid in controlling shipments, inventory and cash. The ELF's cross-sequencing feature could be put to concrete uses, helping to rearrange warehouses (and the company library) by directing the printout of new labels to guide physical rearrangement. Inventories, property numbers and patents could be so catalogued and recatalogued in the ELF. Legal documents, correspondence, company facts and history could be indexed or filed in historical and category trails. And upper management could add private annotations to the public statements, reports and research of both the organization and its competitors, with amendments, qualifications, and inside dope.

PRIDE

While the ELF as described is expected to be general and useful, the original purposes described at the beginning of this paper call for certain further provisions. Now I would like to describe a desirable file and information handling language that will meet these needs, called the PRIDE (Personalized Retrieval, Indexing, and Documentation Evolutionary) System. Its purpose is to facilitate the use of an ELF. The system described is not yet implemented, nor even fully specified, but let us speak as though it is.

PRIDE includes the ELF operations. However, for safety and convenience nearly every operation has an inverse. The user must be permitted, given a list of what he has done recently, to undo it. It follows that "destroy" instructions must

fail safe; if given accidentally, they are to be revocable. For safety's sake, it should take several steps to throw a thing away completely. An important option would permit the user to retrace chronologically everything he does on the system.

Most of PRIDE's applications will involve text handling, either as a primary purpose or in the documentation of some other task. Hence a number of features exist for convenient text usage. Text handling commands (for modifying entries) include the equivalents of standard proofreader's marks for insertion, deletion and switching of sections.

Also for text usage and user comfort, there are certain system non-restrictions. There is no practical restriction on the length of an input entry, and it need follow only the most trivial format conventions. In addition, the machine will interrupt any other PRIDE function to receive input text (inspiration mode). It is necessary that entries of unspecified length be acceptable to the system without fuss or warning. PRIDE does not stipulate fixed record lengths, either for input or storage; any such restrictions would have a psychologically cramping effect. There is no reason the system cannot appear to the user to have no fixed or standard unit lengths; the machine's operating units and sections should not concern him.

Ideally, neither the length of entries, the number of lists, or any other parameter of a file is restricted by anything but the absolute size of all memory. This is a difficult requirement for the programmer. Routinely, however, the system should be able to accept entries thousands of characters long, accept hundreds of entries to a list, and accept hundreds of lists in the file. Otherwise, extraneous consideration by the user of whether there's room to add material or try out an offshoot begins to interfere with the system's use.

Although I have avoided discussing the means by which the user sees his file, PRIDE must, of course, have functions and commands for this purpose. For a CRT these include quick lookup schemes[19], preferably with moving menus and means of readily changing the hierarchy of lookup structure; as well as visual cuing and mnemonic formats, including cursor maneuvers, overlays and animated wipes and other transitions. But such glamorous features do not reduce the challenge or worth of working through a line printer, or seeking to make the system useful under a batch-processing monitor.

Many instructions aside from those already mentioned will be needed by the user; particular applications will require such operations as text lookup and integer arithmetic. And surely all the uses of the system have not been anticipated. Hence a subroutining facility is to be available, reaching to assembly language or opening into the machine's other languages. This could be used for processing the file's contents (e.g., numbers or character strings), or for creating more convenient combined operations out of the different operations dealing with file structure, input-output and text.

PRIDE is one possible way to make an ELF, or any evolutionary file structure, useful. PRIDE would be a foreground, free-standing language with the primary mission of handling files and manuscripts, as discussed at the beginning, and secondary applications in ordering and documenting other kinds of complex information. Its major use would presumably be in connection with time-shared display and information systems. But such a language is only one suggestion. Actually, there is not much reason that the ELF could not be made a standard file structure for all purposes; unused capabilities would not intrude, but would still be there if unex-

pectedly wanted. ELF systems could be built into the file capabilities of general utility software. The actual computation involved is relatively trivial, and the ELF could easily be incorporated into I/O routines or data channel languages. Even small-scale hardware implementations are not unthinkable; a control box between a typewriter and a tape recorder, for instance.

All these applications depend, of course, on the system's being actually useful, which is an empirical question. A number of possible applications have been mentioned. But, except as a crutch to man's fallible mind, is there any reason to suppose that the system has any general applicability in principle?

Philosophy

As "philosophy" I want to speak of two major things. First, complex file structures (like the ELF) make possible the creation of complex and significant new media, the hypertext and hyperfilm. Second, evolutionary file structures (like the ELF) make it possible to keep track of things that have been changing, without our awareness, all along. These include the major categories of human thought, which will go on changing.

Systems of paper have grave limitations for either organizing or presenting ideas. A book is never perfectly suited to the reader; one reader is bored, another confused by the same pages. No system of paper-- book or programmed text-- can adapt very far to the interests or needs of a particular reader or student.

However, with the computer-driven display and mass memory, it has become possible to create a new, readable medium, for education and enjoyment, that will let the reader find his level, suit his taste, and find the parts that take on special meaning for him, as instruction or entertainment.

Let me introduce the word "hypertext"***** to mean a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper. It may contain summaries, or maps of its contents and their interrelations; it may contain annotations, additions and footnotes from scholars who have examined it. Let me suggest that such an object and system, properly designed and administered, could have great potential for education, increasing the student's range of choices, his sense of freedom, his motivation, and his intellectual grasp******. Such a system could grow indefinitely, gradually including more and more of the world's written knowledge. However, its internal file structure would have to be built to accept growth, change and complex informational arrangements. The ELF is such a file structure.

Films, sound recordings, and video recordings are also linear strings, basically for mechanical reasons. But these, too, can now be arranged as non-linear systems-- for instance, lattices-- for editing purposes, or for display with different emphasis. (This would naturally require computer control, using the ELF or a related system, and various cartridge or re-recording devices.) The hyperfilm-- a browsable or vari-sequenced movie-- is only one of the possible hypermedia that require our attention.

So much for what we can create afresh with this structure. What about the things that have already been around awhile?

The physical universe is not all that decays. So do abstractions and categories. Human ideas, science, scholarship and language are constantly collapsing

---

***** The sense of "hyper-" used here connotes extension and generality; cf. "hyperspace." The criterion for this prefix is the inability of these objects to be comprised sensibly into linear media, like the text string, or even media of somewhat higher complexity. The ELF is a hyperfile.

****** I will discuss this idea at length elsewhere.

and unfolding. Any field, and the corpus of all fields, is a bundle of relationships subject to all kinds of twists, inversions, involutions and rearrangement: these changes are frequent but unpredictable. Recall that computers, once a branch of mathematics, are now their own field (but the development of fluid logic indicates a possible merger with the art of wind instruments). Social relations, psycholinguistics and psychonomics are new fields, even though they rest on no special discoveries; political economy, natural history and social ethics are gone. Within a given area, too, the subheadings of importance are in constant flux. In the social sciences, for instance, the topic headings of the nineteen-thirties now sound quaint.

While the disappearance and up-ending of categories and subjects may be erratic, it never stops; and the meaning of this for information retrieval should be clear. Last week's categories, perhaps last night's field, may be gone today. To the extent that information retrieval is concerned with seeking *true* or *ideal* or *permanent* codes and categories-- and even the most sophisticated "role indicator" syntaxes are a form of this endeavor-- to this extent, information retrieval seems to me to be fundamentally mistaken. *The* categories are chimerical (or temporal) and our categorization systems must evolve as they do. Information systems must have *built in* the capacity to accept the new categorization systems as they evolve from, or outside, the framework of the old. Not just the new material, but the capacity for new arrangements and indefinite rearrangements of the old, must be possible. In this light, the ELF, indefinitely revisible and unperturbed by changes in overall structural relations, offers some promise.

There is, then, a general rationale. I believe that such a system as the ELF actually ties in better than anything previously used with the actual processes by which thought is progressively organized, whether into stories or hypertext or library categories. Thus it may help integrate, for human understanding, bodies of material so diversely connected that they could not be untangled by the unaided mind. For both logistic and psychological reasons it should be an important adjunct to imaginative, integrating and creative enterprises. It is useful where relationships are unclear; where contingencies and tasks are undefined and unpredictable; where the structures or final outcome it must represent are not yet fully known; where we do not know the file's ultimate arrangement; where we do not know what parts of the file are most important; or where things are in permanent and unpredictable flux. Perhaps this includes more places than we think. And perhaps here, as in biology, the only ultimate structure is change itself.

CONCLUSION

This paper has proposed a different kind of structure for handling information.

Essentially it is a file with certain storage provisions which, combined, permit the file's contents to be arranged any-which-way, and in any number of ways at once. A set of manipulation functions permits making changes or keeping track of developments. The file is capable of maintaining many different arrangements at the same time, many of which may be dormant. This makes ordinary measures of efficiency inappropriate; as with high fidelity music systems, enrichment is derived from the lavish use of surplus capacity.

The key ideas of the system are the inter-linking of different lists, regardless of sequence or additions; the re-configurable character of a list complex into any humanly conceivable forms; and the ability to make copies of a whole list, or

list complex-- in proliferation, at will-- to record its sequence, contents or arrangement at a given moment. The Evolutionary List File is a member of the class of evolutionary file structures; and its particular advantages are thought to be psychological, not technical. Despite this file's adaptability to complex purposes, it has the advantage of being conceptually very simple. Its structure is complete, closed, and unified as a concept. This is its psychological virtue. Its use can be easily taught to people who do not understand computers. We can use it to try out combinations that interest us, to make alternatives clear in their details and relationships, to keep track of developments as they occur, to sketch things we know, like or currently require; and it will stand by for modifications. It can be extended for all sorts of purposes, and implemented or incorporated in any programming language.

There are probably various possible file structures that will be useful in aiding creative thought. This one operates, as it were, on lists that hook together sideways, and their copies. There may be many more.

REFERENCES

1. Bush, Vannevar, "As We May Think." *The Atlantic Monthly*, 176:1 (July, 1945), 101-108.
2. Hirsch, Phil, "The Bush Rapid Selector." *Datamation*, June 1965, 56-7.
3. Wallace, Everett M., "Experience with EDP Support of Individuals' File Maintenance." *Parameters of Information Science: Proceedimgs of the American Documentation Institute*, v.I (American Documentation Institute, 1964), 259-261.
4. Gorey, Edward, *The Unstrung Harp; or, Mr Earbrass Writes a Novel*. Duell, Sloan and Pearce, N.Y., 1953.
5. International Business Machines Data Processing Division, "The IBM Administrative Terminal System." Company brochure 520-1146.
6. International Business Machines Technical Publications Department, "1440-/1460 Administrative Terminal System Application Description." IBM, White Plains, New York.
7. Timberlake, W.D., "Administrative Terminal System" (abstract.) *STWP Proceedings*, May 1965, no page number. New York: Society of Technical Writers and Publishers, 1965.
8. Farrell, Austin C., "Evolution of Automated Writing." *STWP Proceedings*, May 1965, no page numbers. New York: Society of Technical Writers and Publishers, 1965.
9. Bobrow, Daniel G. and Bertram Raphael, "A Comparison of List-Processing Languages." *Comm. ACM* 7:4 (April, 1964), 231-240.
10. Comfort, W.T., "Multiword List Items." *Comm. ACM* 7:6 (June, 1964), 357-362.
11. Weizenbaum, J., "Knotted List Structures." *Comm. ACM* 5:3 (March, 1962), 161-165.
12. Prywes, N.S. and H.J. Gray, "The Multilist System for Real Time Storage and Retrieval." *Proceedings of IFIP Conference*, 1962, 112-116.
13. Prywes, N.S., "Interim Technical Report: The Organization of Files for Command and Control." Moore School of Engineering, March, 1964.
14. Bachman, C.W. and S.B. Williams, "The Integrated Data Store-- A General Purpose Programming System for Random Access Memories." *AFIPS Conference Proceedings*, v. 26, 411-422. Spartan Books, 1964.
15. General Electric Computer Department, "I-D-S." Company brochure CPB-425.
16. General Electric Computer Department, "Introduction to Integrated Data Store." Company brochure CPB-1048, 1965.
17. Bachman, Charles W., "Software for Random Access Processing." *Datamation*, April 1965.
18. General Electric Computer Department, "Integrated Data Store: New General Electric Technique for Organizing Business Data." Company publication, January 1965.
19. Corbin, Harold S., and George J. Stock, "On-Line Querying via a Display Console." *Fourth National Symposium on Information Display: Technical Session Proceedings*, 127-154. Society for Information Display, Washington, 1964.

list complex-- in proliferation, at will-- to record its sequence, contents or arrangement at a given moment. The Evolutionary List File is a member of the class of evolutionary file structures; and its particular advantages are thought to be psychological, not technical. Despite this file's adaptability to complex purposes, it has the advantage of being conceptually very simple. Its structure is complete, closed, and unified as a concept. This is its psychological virtue. Its use can be easily taught to people who do not understand computers. We can use it to try out combinations that interest us, to make alternatives clear in their details and relationships, to keep track of developments as they occur, to sketch things we know, like or currently require; and it will stand by for modifications. It can be extended for all sorts of purposes, and implemented or incorporated in any programming language.

There are probably various possible file structures that will be useful in aiding creative thought. This one operates, as it were, on lists that hook together sideways, and their copies. There may be many more.

REFERENCES

1. Bush, Vannevar, "As We May Think." *The Atlantic Monthly*, 176:1 (July, 1945), 101-108.
2. Hirsch, Phil, "The Bush Rapid Selector." *Datamation*, June 1965, 56-7.
3. Wallace, Everett M., "Experience with EDP Support of Individuals' File Maintenance." *Parameters of Information Science: Proceedings of the American Documentation Institute*, v.1 (American Documentation Institute, 1964), 259-261.
4. Gorey, Edward, *The Unstrung Harp; or, Mr Earbrass Writes a Novel*. Duell, Sloan and Pearce, N.Y., 1953.
5. International Business Machines Data Processing Division, "The IBM Administrative Terminal System." Company brochure 520-1146.
6. International Business Machines Technical Publications Department, "1440-/1460 Administrative Terminal System Application Description." IBM, White Plains, New York.
7. Timberlake, W.D., "Administrative Terminal System" (abstract.) *STWP Proceedings*, May 1965, no page number. New York: Society of Technical Writers and Publishers, 1965.
8. Farrell, Austin C., "Evolution of Automated Writing." *STWP Proceedings*, May 1965, no page numbers. New York: Society of Technical Writers and Publishers, 1965.
9. Bobrow, Daniel G. and Bertram Raphael, "A Comparison of List-Processing Languages." *Comm. ACM* 7:4 (April, 1964), 231-240.
10. Comfort, W.T., "Multiword List Items." *Comm. ACM* 7:6 (June, 1964), 357-362.
11. Weizenbaum, J., "Knotted List Structures." *Comm. ACM* 5:3 (March, 1962), 161-165.
12. Prywes, N.S. and H.J. Gray, "The Multilist System for Real Time Storage and Retrieval." *Proceedings of IFIP Conference*, 1962, 112-116.
13. Prywes, N.S., "Interim Technical Report: The Organization of Files for Command and Control." Moore School of Engineering, March, 1964.
14. Bachman, C.W. and S.B. Williams, "The Integrated Data Store-- A General Purpose Programming System for Random Access Memories." *AFIPS Conference Proceedings*, v. 26, 411-422. Spartan Books, 1964.
15. General Electric Computer Department, "I-D-S." Company brochure CPB-425.
16. General Electric Computer Department, "Introduction to Integrated Data Store." Company brochure CPB-1048, 1965.
17. Bachman, Charles W., "Software for Random Access Processing." *Datamation*, April 1965.
18. General Electric Computer Department, "Integrated Data Store: New General Electric Technique for Organizing Business Data." Company publication, January 1965.
19. Corbin, Harold S., and George J. Stock, "On-Line Querying via a Display Console." *Fourth National Symposium on Information Display: Technical Session Proceedings*, 127-134. Society for Information Display, Washington, 1964.

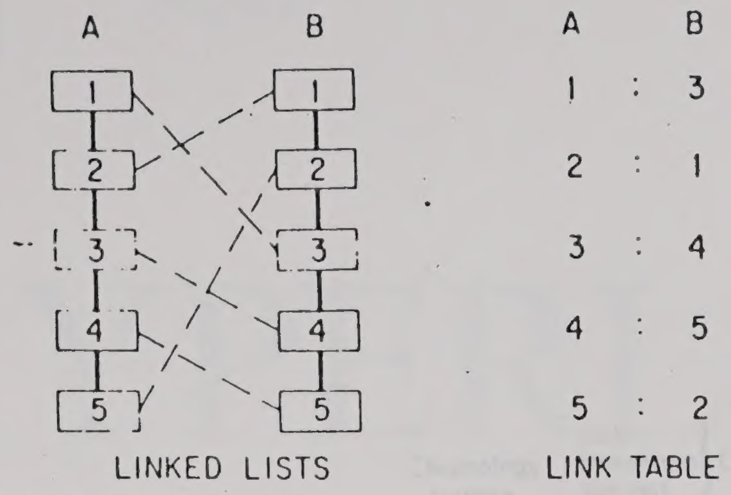

A
B
1
2
3
4
5
1
2
3
4
5
LINKED LISTS
A B
1 : 3
2 : 1
3 : 4
4 : 5
5 : 2
LINK TABLE

FIGURE 1—Zippered lists: 1-for-1 links between entries are invariant under list permutation.

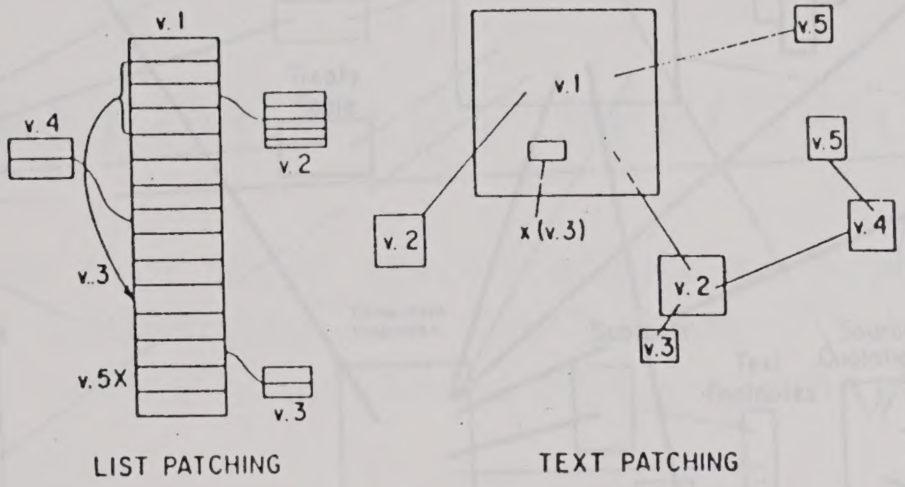

v. 1
v. 4
v. 2
v. 3
v. 5X
v. 3
LIST PATCHING
v. 1
v. 5
x (v. 3)
v. 2
v. 5
v. 4
v. 2
v. 3
TEXT PATCHING

FIGURE 2—Spinoff of variants: extra versions need little space.

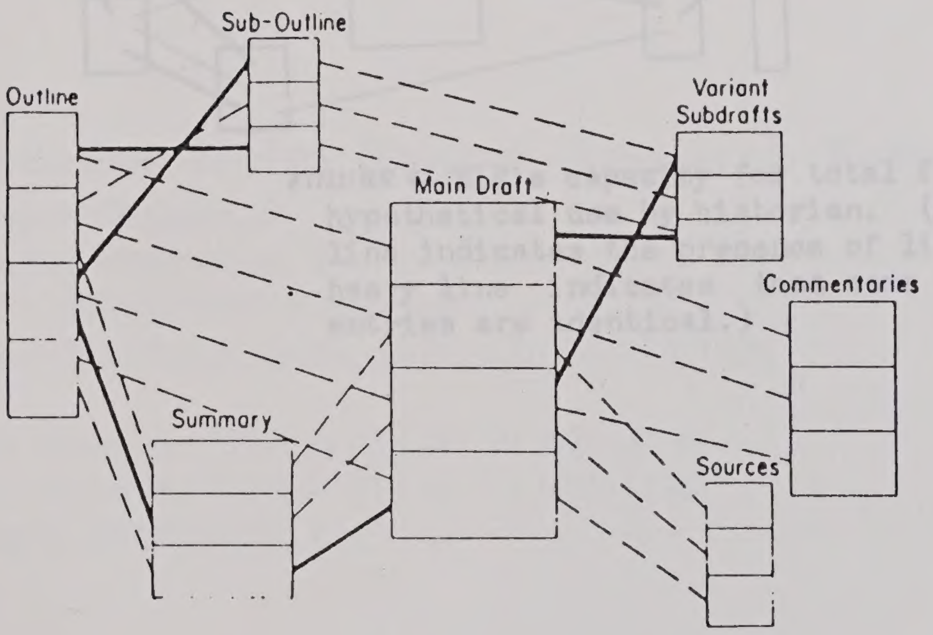

Sub-Outline
Outline
Variant Subdrafts
Main Draft
Commentaries
Summary
Sources

FIGURE 3— All levels of documentation may be contained in the ELF. (Heavy lines indicate that linked entries are identical.)

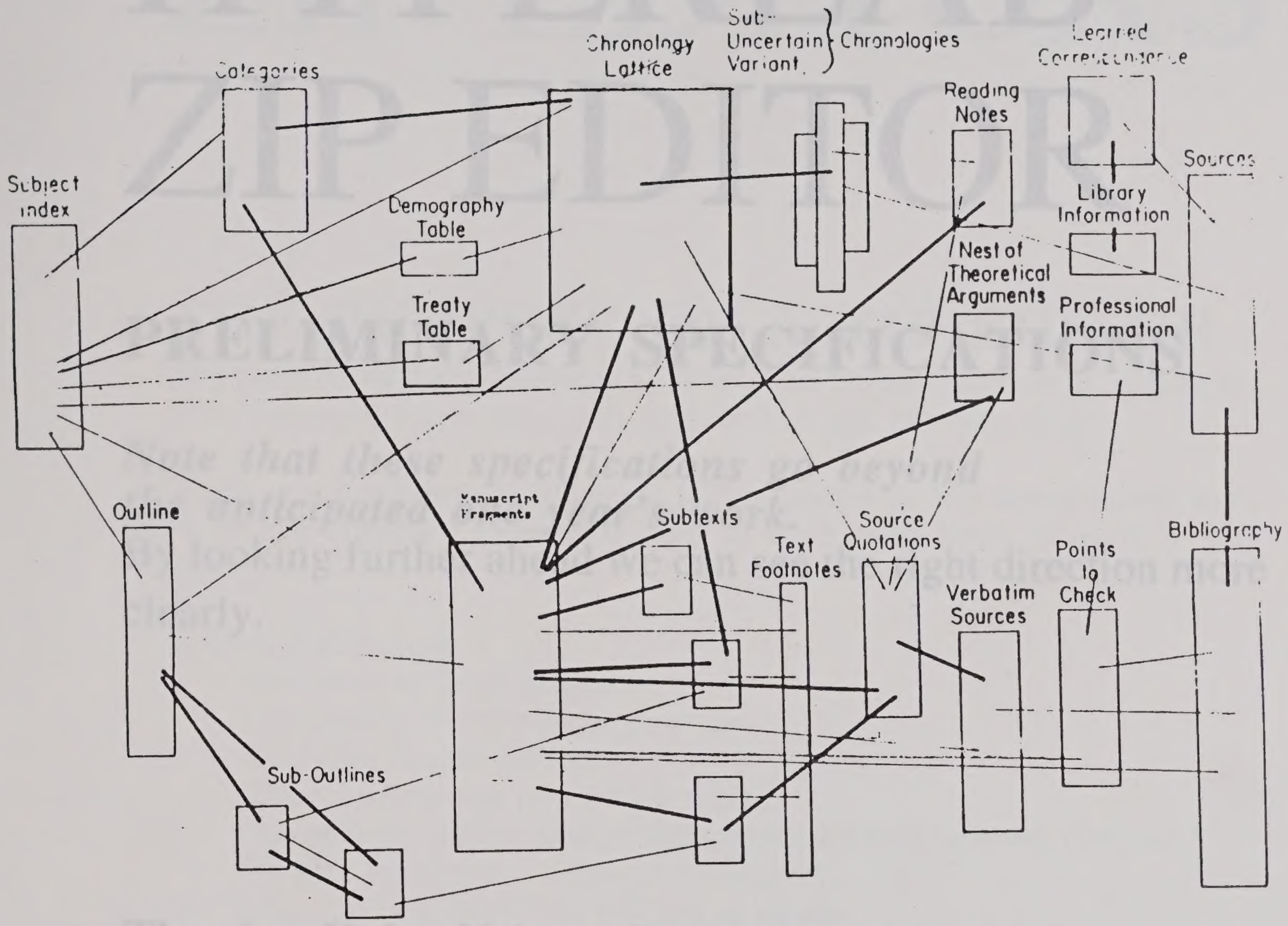

Subject index
Categories
Chronology Lattice
Sub-
Uncertain
Variant
Chronologies
Learned Correspondence
Reading Notes
Sources
Library Information
Demography Table
Treaty Table
Nest of Theoretical Arguments
Professional Information
Outline
Manuscript Fragments
Subtexts
Text Footnotes
Source Quotations
Points to Check
Verbatim Sources
Bibliography
Sub-Outlines

***FIGURE 4*— ELF's capacity for total filing: hypothetical use by historian. (A thin line indicates the presence of links; a heavy line indicates that some linked entries are identical.)**

HYPERLAB ZIP EDITOR

PRELIMINARY SPECIFICATIONS

Note that these specifications go beyond the anticipated one year's work.
By looking further ahead we can see the right direction more clearly.

Theodor Holm Nelson, Research Fellow
Sapporo HyperLab
Doshin Maruyama, 8F
2-20, Odori Nishi 20-chome
Chuo-Ku, Sapporo-shi, Hokkaido 064 JAPAN

Tel. (from the States) 011 81 11 621-1133
Fax (from the States) 011 81 11 621-1170
e-mail ted@xanadu.net

ZIP EDITORS

THE PURPOSE OF A ZIP EDITOR

The purpose of a Zip Editor is to show, and let you work with, connected data sequences (especially text), and the linkages between them.

Using our Zip Editor, we hope it will be easier to edit conventional text, to handle complex hypertexts, and to work with large projects of any kind.

Special uses include:
- Outlining of large projects
- Organizing of very large projects
- Evolving Sequential text
- Cross-referencing of Everyday lists
  - (especially lists of things to do, calendar lists, address lists, etc.)
- Parallel hypertext (collateral hypertext)

and others that we hope you will tell us about.

Perhaps most important, the Zip Editor is a first simple step toward transclusive publishing, where authors may re-use each others' materials without restriction but with reward and clarity.

BASIC OBJECTS

Item

An item is a chunk, paragraph-like object, or file card-like object.
(It is stored as a single Unix file. The file's name is not known to the user.)

Subitem

An item may have several subitems or parts (sometimes called "fields."
These include Title, Body, Date.
(If an item has Categories or Keywords, those are translated into Zip Addresses.)

Zip

A zip is a series of items, with or without a name.
The items are in sequence.
A zip may be used--
    to store contents
    to select or filter other zips.

Head

Main item, usually at the top, usually the title.

Tail, Staff, Shaft, Pole

Vertical line on which items are mapped.

Wing, Window

Enlarged view of items, or part of items, extending from the Tail.

Zipplex

A zipplex is a set of zips which a user wants to consider together. It may be a collection of projects, a single project, or part of a project. (Internally, it is a list of zips.)

Segment

Series of items separated by an operation.

Gap

Logical break between segments, where a transclusion is not shown.

Phantom Zip, Projected Zip

A phantom, or projected, zip is a zip the user asks to see which is not actually "created." (Technically, it is considered to be a view rather than a construction. Even though the zip is actually constructed-- and demolished if not kept, its existence is not recorded in structure history unless the user decides to keep it.)

Journal

This is a zip which records all new items, in order.

Ray

A ray is a line or stripe between items on two different zips, to represent transclusion or linkage. Rays may be of different colors, and may be animated.

History

The History is the record of what operations have been performed.
(Unlike most recorded series of operations, which are sequential and must be played back sequentially, our history is tree-structured; you may go back to any previous state of the system, or of a given zip, and perform operations from that point. Thus history is a tree structure.

Structure history tells what constructions were actually built.

View history is much richer, and tells what views were actually seen.

Control Pad

These are the different controls of the system, being mainly buttons and selector pads.

Selector Pad

Pad which can be peeled off and placed over items and zips to be selected.

Menuplex

The Menuplex is a set of selector pads, collected together. They show the possible operations symbolically, rather than in words.

Muster

Basic state where all zips can be seen and reviewed.

BASIC RELATIONS

Connection

Either a transclusion (see below) or a link (see below).

Transclusion

A transclusion is the appearance of the same item in more than one zip. (Technically, it is not copied; rather, the name of the item is included in the zip, which is a list.)

Link

A link is a connection between items on different zips which is not a transclusion.

Sequence

Sequence is the relation between successive items on a single zip.

Succession

When an item is edited, a new item is created.
Items may be split or combined.
Each item knows its previous and later versions.

BASIC VIEWS

Seahorse View

View of a zip showing flag at top, vertical bar, possibly windows on the vertical bar.

Pole, Shank or Staff

The vertical portion of a displayed zip.

Head

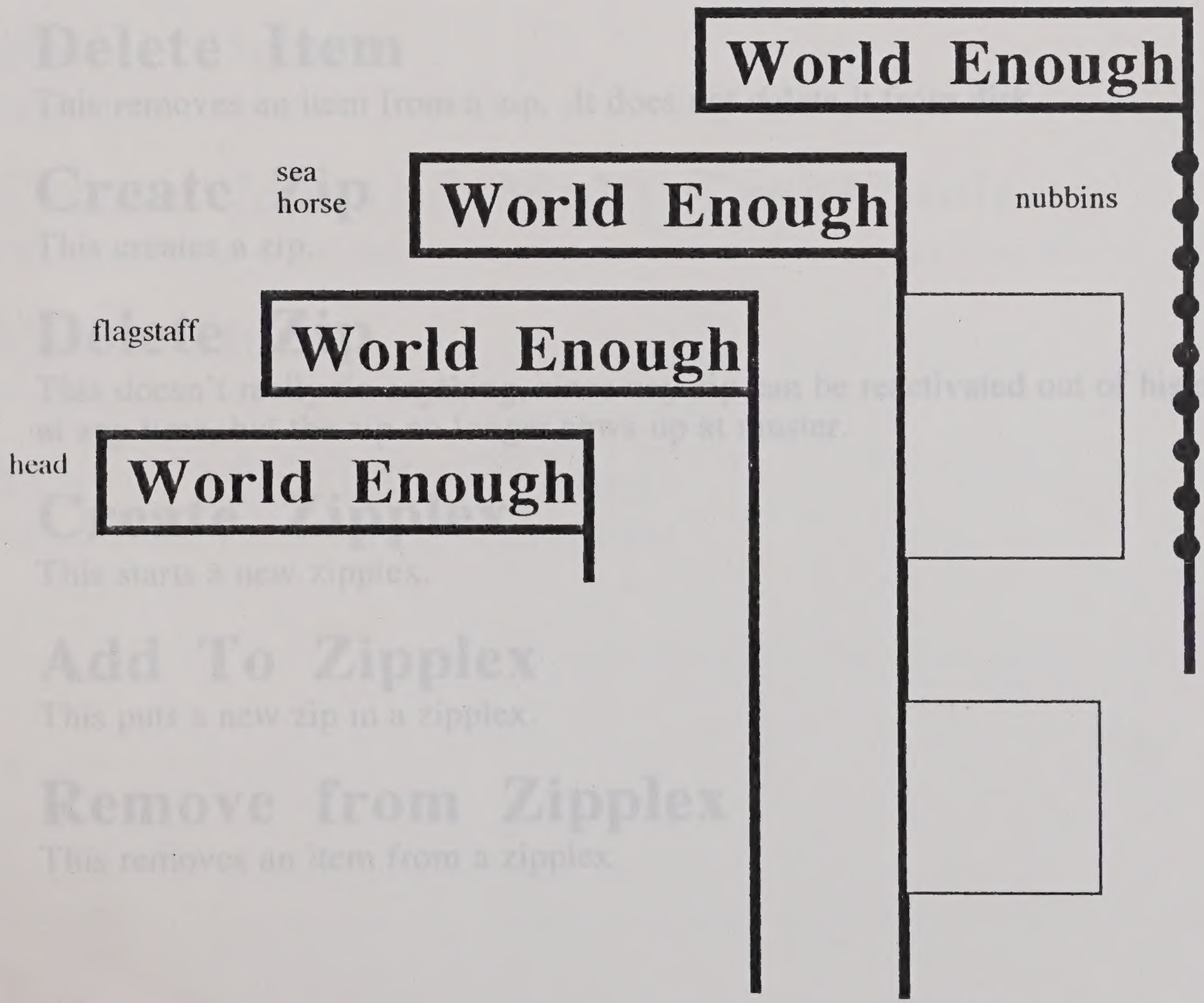
The top part of a displayed zip, showing the title.

Window

The flaglike rectangle in which contents of a Zip are shown.

Nubbin or Nub

Slight swelling on the staff of a zip to indicate an item.


World Enough
sea horse
World Enough
nubbins
flagstaff
World Enough
head
World Enough

BASIC OPS: Structure Ops

Create Item

This creates a new item item on some particular zip.

Transclude Item

This adds an existing item to an existing zip.

Edit Item

This creates a new item which replaces the previous.

Delete Item

This removes an item from a zip. It does not delete it from disk.

Create Zip

This creates a zip.

Delete Zip

This doesn't really do anything, since any zip can be reactivated out of history at any time, but the zip no longer shws up at muster.

Create Zipplex

This starts a new zipplex.

Add To Zipplex

This puts a new zip in a zipplex.

Remove from Zipplex

This removes an item from a zipplex.

BASIC OPS: Compound Structure Ops

Multi Transclude

This transcludes multiple items between zips.

Multi Delete

This deletes a number of items from an existing zip.

Create Zip and Multi Transclude

This creates a new zip and transcludes a number of items into it, or all.

Move Item

Moves an item up or down a zip.

Move Multiple Items

Moves a selection of items up or down a zip.

BASIC OPS: View Ops

Scroll in zip

The material scrolls through a window, and the window moves down the zip's pole.

Reposition Zip(s)

One or more selected zips are moved about on the screen.

Resize Zip

The appearance of a zip is made larger or smaller.

Open Wing/Window of Zip

A selected portion of the Zip is turned from a vertical bar into an open window on that bar.

Close Wing/Window of Zip

The window is closed.

Resize window (text size constant)

The text window gets larger or smaller, text stays same size.

Resize window (text size changing)

The window is resized and the text gets bigger or smaller.

Scroll in zip

The material scrolls through a window, and the window moves down the zip's pole.

Show Links

This shows links between two or more zips.

Show Transclusions

This shows transclusions between two or more zips.

BASIC OPS: Compound View/Structure Ops

Select Zipplex

Only certain zipplex(es) are shown.

Filter/Cross-Select

Selections in one zip are made to specifications from other zips.

Multi Select (items)

A number of items are selected.

Select Transclusions

At least three zips are involved. One, a selector, choses which transclusions ar to be shown betwen the other two.

Select Segments

Using transclusions specified by one zip, segments are selected.

Select Linked Items/Nonlinked Items

The equivalent of select transclusions and select items, but for links rather than transclusions.

(Compound Selections)

Segments and/our transclusions and/or linked material are selected based on criteria in several filter or selector zips.

Make/(Project Phantom)

One or more compound zips are made or projected for study, and their contents selected.

Ratify Phantom

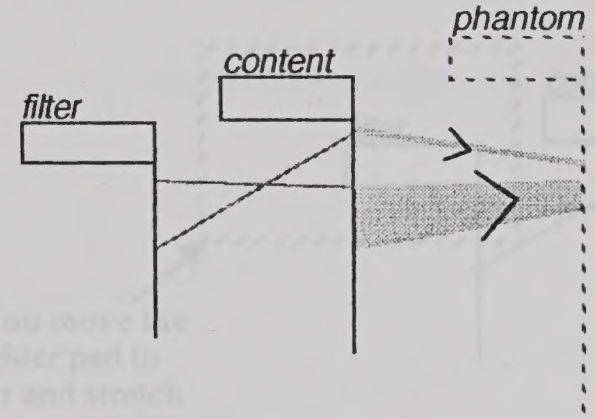
In case you like the phantom zip(s) you just made, you decide to keep them. They now become official, and their creation goes into structure history.

COMPOUND OP VIEWS

Filtered Views

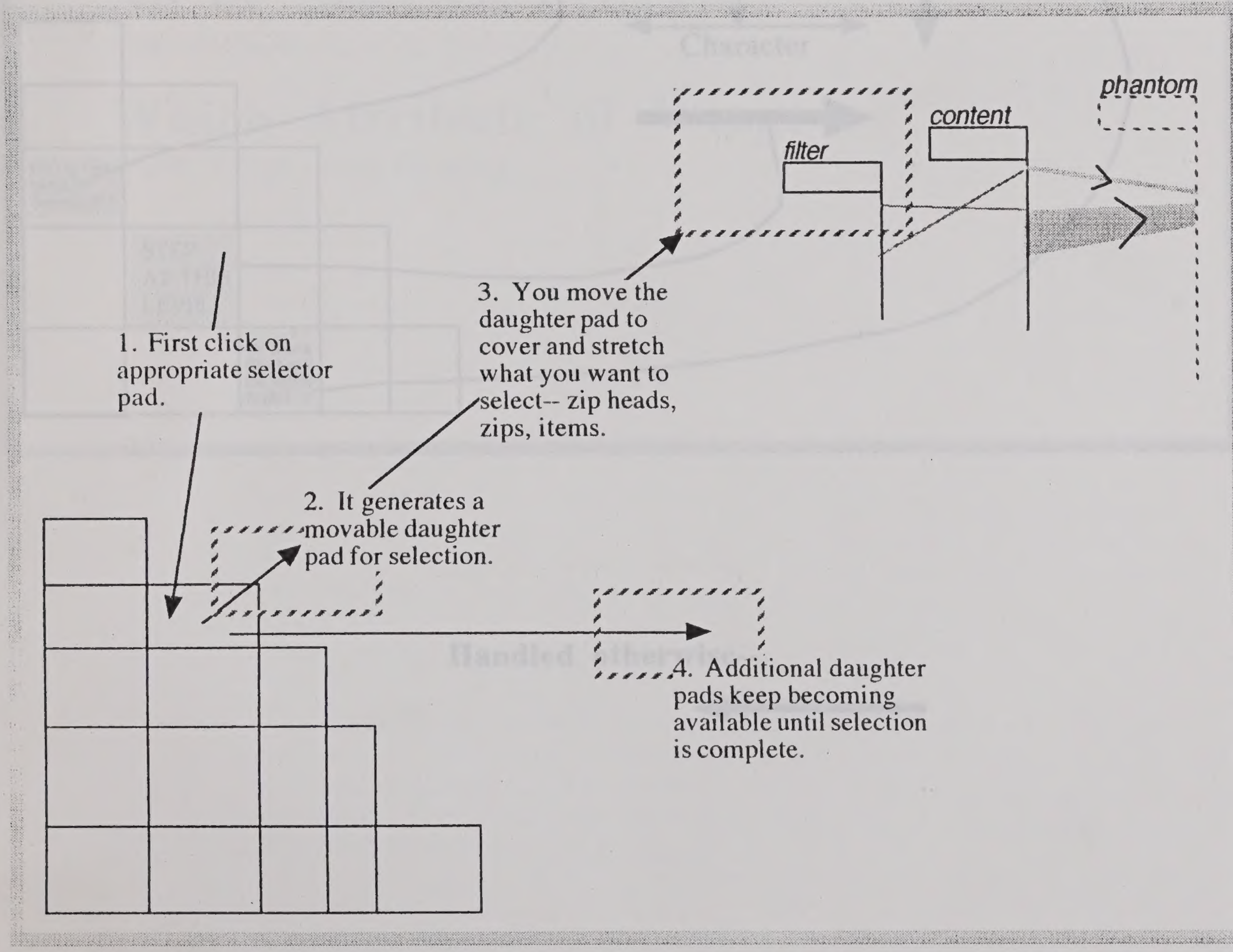
One Zip can filter what shows on another zip, or what is projected from some other zip to a new zip.

There will be a wide variety of such functions.



BASIC CONTROL LAYOUT

1. GRAPHIC SELECTION METHOD



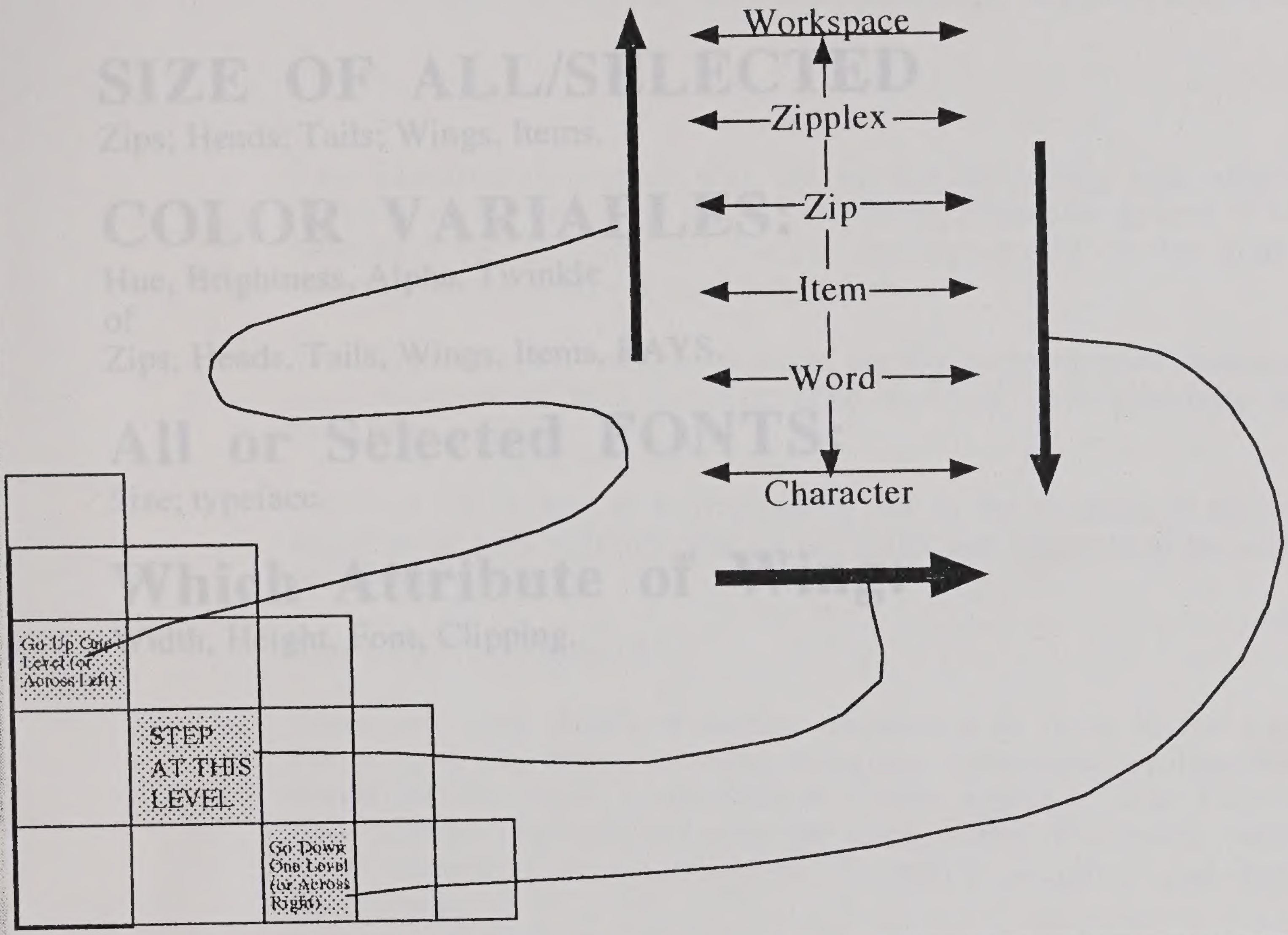
1. First click on appropriate selector pad.

2. It generates a movable daughter pad for selection.

3. You move the daughter pad to cover and stretch what you want to select-- zip heads, zips, items.

4. Additional daughter pads keep becoming available until selection is complete.

2. SEQUENTIAL SELECTION METHOD (Ringstep)


Workspace
Zipplex
Zip
Item
Word
Character
Go Up One Level (or Across Left)
STEP AT THIS LEVEL
Go Down One Level (or Across Right)

Other Uses of Ringstep

SIZE OF ALL/SELECTED
Zips, Heads, Tails, Wings, Items.

COLOR VARIABLES:
Hue, Brightness, Alpha, Twinkle
of
Zips, Heads, Tails, Wings, Items, RAYS.

All or Selected FONTS:
Size, typeface.

Which Attribute of Wing:
Width, Height, Font, Clipping.

Handled otherwise--

Other Uses of Ringstep

SIZE OF ALL/SELECTED

Zips; Heads; Tails; Wings, Items.

COLOR VARIABLES:

Hue, Brightness, Alpha, Twinkle
of
Zips, Heads, Tails, Wings, Items, RAYS.

All or Selected FONTS:

Size; typeface.

Which Attribute of Wing:

Width, Height, Font, Clipping.

HyperLab as a Human Media System

Marlene Mallicoat, Research Associate, Sapporo HyperLab

The Sapporo HyperLab was set up for the study and creative development of computer media. However, it has also proven to be a laboratory for the study and creative development of another kind of information medium-- human interaction.

The HyperLab has brought together the Japanese spirit of teamwork and cooperation and the American style of individualistic and imaginative energy.

I believe the HyperLab is flourishing due to the merging of the best qualities of both cultures, and of the skills and interests of its varied members.

The Lab Environment

There are many kinds of human interaction to study in the Lab-- many subjects, tasks, puzzles, surprises, questions, explanations, investigations, jokes, conversations-- from Apples to Zips, Ethernet and Internet, Digicash and software crash. Deep discussions among team members are punctuated by much laughter and light-heartedness.

It's not easy to achieve the kind of harmony and productivity that characterize the Lab, but that harmony has been naturally easier in the spacious and pleasant surroundings that have been so generously provided-- a cheery and well-managed setting, thoughtfully watched over by SEC and our able office manager.

There are many long hours at the screen

The working day sometimes lasts from lunch to dawn ("It's a real Lab now!" said Ted, when some of the other members started working until dawn along with him.) We are grateful for the all-night 7-11 store downstairs, and the non-smoking environment which the members so considerately maintain.

The Projects

There are many background concerns, like networking the mix of UNIX workstations, IBM PCs and Apple Macintoshes that showed up at the Lab, or our constant study of the World Wide Web and its continuing developments.

But the center of the work is the deep and visionary ambition shared by both IntelligentPad and Xanadu, both built around the hope of empowerment and a better life for everyone. Today's Internet provides an astonishing new environment that can help bring these dreams to life.

The People

While the Lab members represent their individual companies with considerable devotion, as a team they share a great camaraderie and enthusiasm for Lab projects.

A remarkable aspect of the Lab is the high level of technical skill, and particular type of cleverness, that each team member brings to the project. We come from a variety of backgrounds: chemistry, precision engineering, mathematics, philosophy and film-making; but working together unifies us all.

Professor Tanaka's gentle guidance combines both traditional Japanese leadership qualities and administrative skills with an American-style imagination. He brings out the best in everybody. Ted Nelson, on the other hand, combines his American rush of kinetic and far-seeing inspirations, with a Japanese perfectionism in his writing and a Japanese level of caring about what he is working on.

One Suggestion

Perhaps the one most significant improvement every team member could make would be to learn more of the other's language, thus allowing for even deeper and more enriching communications

Zip Editor

大久保清貴

1995年3月22日

1　情報の構造とトランスクルージョン

情報の変化は、新しいものが生まれることによるよりも、複数の情報が合成されたり、順序が入れ替わったり、削除されることによる割合がずっと多い。典型的な例としてうわさ話が伝搬する様子を見てみよう。

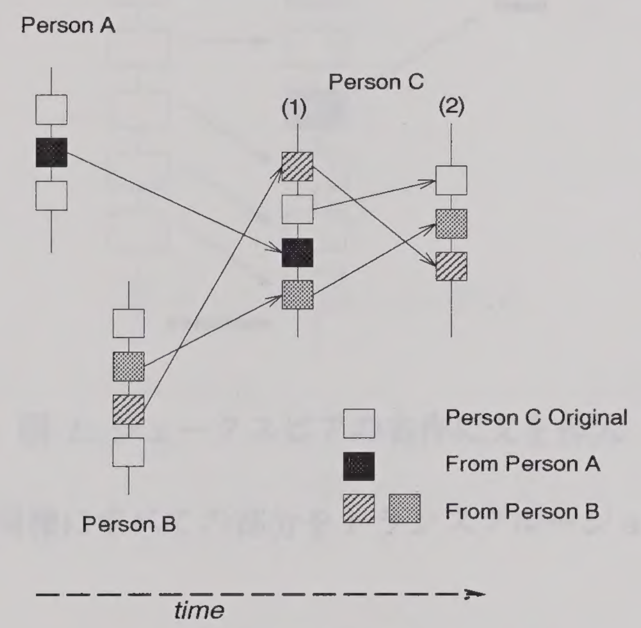

Person A
Person C
(1)
(2)
Person B
Person C Original
From Person A
From Person B
time

図1: うわさ話の伝搬

ここでは、Aさんの話の一部がCさんに伝わり、Bさんの話の一部も合成される。図1の(1)の状態ではBさんの話の順番が入れ替わり、Cさん自体の解釈も組み込まれている。時間がたって(2)の状態になると話の順番が再び入れ替わり、一部は消えている。これが人間が自然に行ってきた知識の編集の基本的な形である。

World Wide Web(WWW)を使っていると、その膨大な情報を自分の興味にしたがって体系化したいと思うようになる。しかしWWWにはナビゲーション・リンクしかないので、得た情報をあたかも自分のものであるかのように、自由に扱えない。例えば、あるテーマで文章を書くために情報を収集しているときに、他の人が同じような情報コレクションをHTMLページにまとめているのを見つけることがある。しかし、このページを直接書き変えることは、他人の所有物であるからできない。また、このページを持ってきて一部を修正することはできるが、それを自分のページとして他のサイトからアクセスできる場所に置くのは、たとえ著作権にふれないとしても、ためらわれる。

ネットワーク時代になって、情報を収集することはたやすくなったが、ここまでは他人のアイデアだが、ここは自分の独創であるということを主張する根本的な方法がないために、いざ外に向けて出版するときに困ったことになる。ネットワークにより他人もたやすくこちらの出版したものを見つけられるので、他人のアイデアを含む形の出版をした場合、たちまちクレームをつけられることになるだろう。

この問題を Ted Nelson 氏は1960年代からすでに気がついて [1] Xanadu[1]をライフワークとしてきた。上に述べた知的貢献部分の判別、そして著作権侵害の心配をせずに、他人の知的成果の上に自分の創作を加えられるしくみがトランスクルージョンである。

ナビゲーション・リンクとトランスクルージョンを混同している人がよくいる。トランスクルージョンを使えば「シェークスピアの名作に文を挿入」(図2) することができる。まず新しい縦棒をひとつもってきて、原文をすべてトランスクルードする。そして好みの個所に文を挿入すればよい。トランスクルージョンを支えるメカニズムがすべての部分の原著者をトラッキングしているからである。

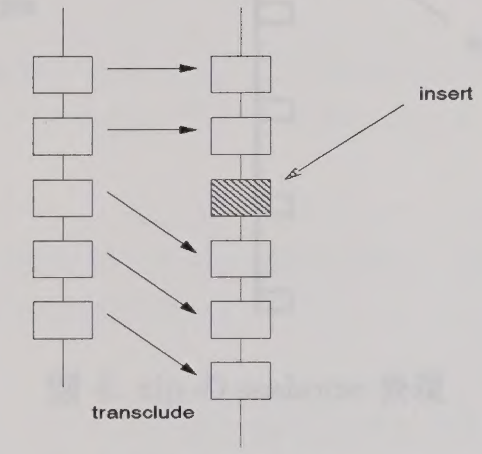


図 2: シェークスピアの名作に文を挿入

またパロディを書く場合も同様にすべての部分をトランスクルージョンして適当な個所の筋を代えればよい。

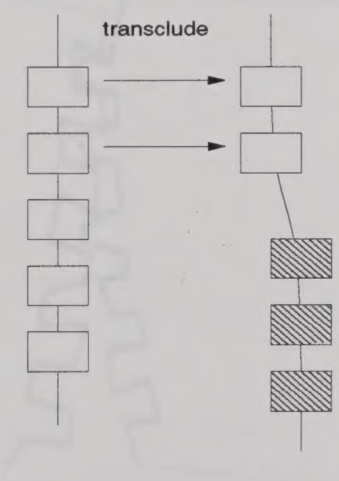


図 3: パロディの創作

また研究論文の場合もそうである。研究論文では、たまにビジョナリーな人が他の研究と比類できないものを書く場合があるが、多くは他の研究との関連を十分付けてコンテキストをもたせるの普通である。

---

[1] Xanadu は Project Xanadu の登録商標である。

2　Zip Editor とは？

Zip Editor は、トランスクルージョンを編集するエディタである。前節の図で縦棒が Zip にあたる。Ted Nelson 氏はこのエディタを、個人の創造性を自由に発揮させるためのエディタと位置づけている [2]。Zip エディタは、ネットワーク・パブリッシング・システム Xanadu のひとつのコンポーネントでもある。

Zip は項目の集まりから構成される。例えばある出来事のメモ、調査中の資料、スケジュールの日程などを表すために使う。Zip のひとつの視覚表現として図 4のような「竜の落子」(seahorse) [3] を用いる。

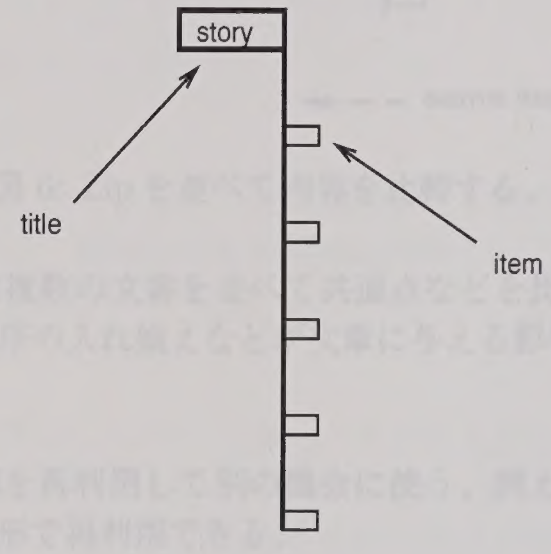


図 4: zip の seahorse 表現

Zip の名前は図 5のようなジッパーから由来している。ジッパーが布を結びつけるように、Zip エディタは図 6のように Zip を両側に並べて内容を比較するのに便利である (collateral hypertext)。図 6の広がった部分は、項目がオープンされて内容がもっと見えるようになった状態を表す。点線は同じ項目を再利用していることを示している。

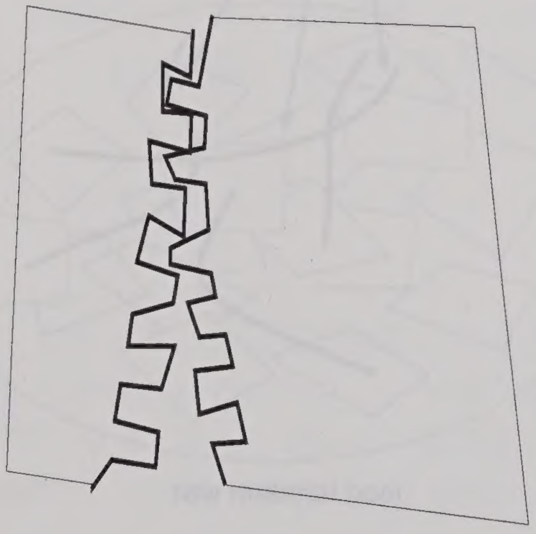


図 5: チャックを締めるジッパー

Zip による内容の比較と再利用はユーザにとって次のような意味がある。

- 比較 (inter-comparison)

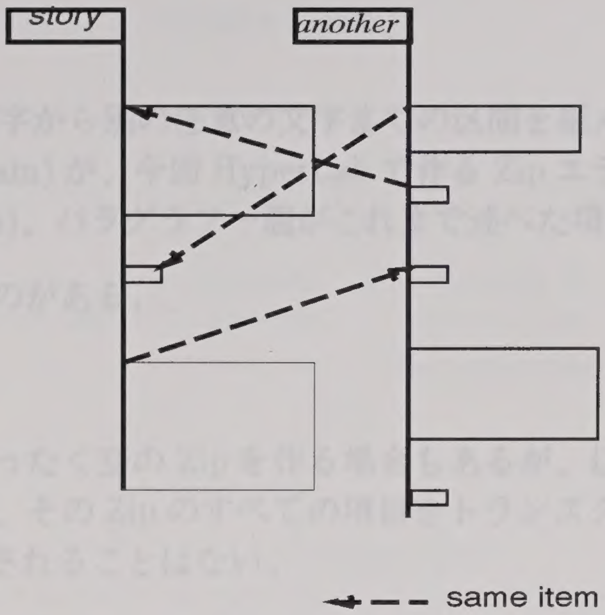

story
another
same item

図 6: Zip を並べて内容を比較する。

特定の出来事を記述した複数の文書を並べて共通点などを比較するために使う。また文書を書く場合に、項目の順序の入れ換えなどが文章に与える影響を調べるのに使う。また、、。

- 再利用 (reuse)

  以前に書いた文章の一部を再利用して別の機会に使う。例えば、条例が改定された場合、変更個所が分かるような形で再利用できる。

Zip は、文書のデータベースからストーリーをつむぎ出す糸であるともいえる (図 7)。

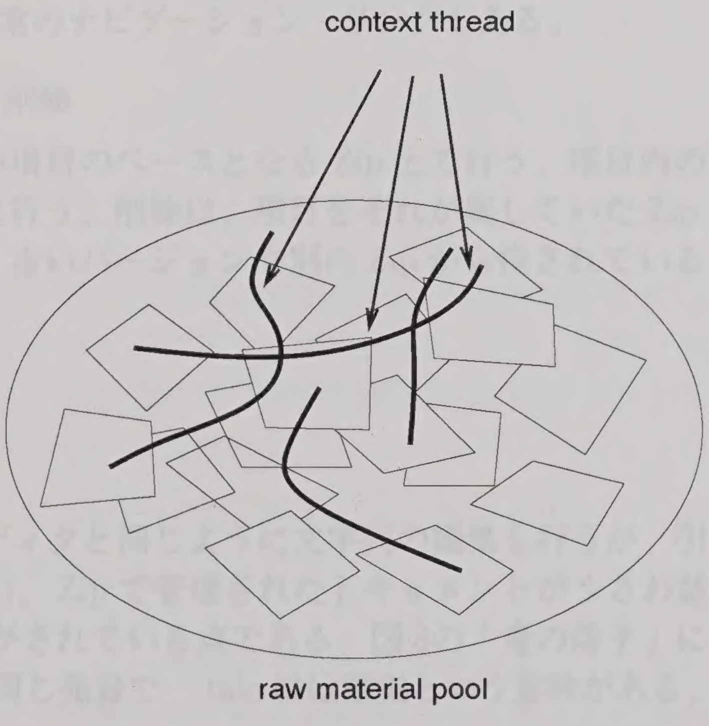

context thread
raw material pool

図 7: zip は素材を再利用する。

3　Zip の操作

Xanadu では文章の任意の文字から別の任意の文字までの区間を選んでトランスクルージョンすることになっている (fine grain) が、今回 HyperLab で作る Zip エディタはパラグラフをひとつの塊として扱う (coarse grain)。パラグラフ一個がこれまで述べた項目にあたる。

Zip の操作には次のようなものがある。

- Zip の作成

  新しい Zip を作る。まったく空の Zip を作る場合もあるが、以前に作った Zip をバージョン・アップするために、その Zip のすべての項目をトランスクルードして始めることもできる。 Zip は通常削除されることはない。

- 並べ変え (rearrangement)

  項目の並べ変えを行う。このとき Zip の変更履歴にこの操作が記録される。この記録により、以前の状態に戻ったり、編集操作を視覚的に再現したりできる。並べ変える項目の選択、挿入される位置の指定にはマウスによるダイレクト・マニピューレーションを用いる。

- トランスクルード

  トランスクルードするときは、トランスクルードされる方とする方の両方の Zip を画面上に表示して、並べ変えと同様にダイレクト・マニピューレーションで行う。この操作も Zip の変更履歴に記録される。トランスクルードされた項目の削除は、通常の項目の削除と同様に行われる。

- リンク

  Zip エディタには通常のナビゲーション・リンクもある。

- 項目の作成／変更／削除

  項目の作成は、その項目のベースとなる Zip 上で行う。項目内の変更は通常のワードプロセッサと同じように行う。削除は、項目をそれが属していた Zip 上からはずすことで行われる。削除しても、古いバージョンや別の Zip から指されているので、ディスク上から消えることはない。

4　Zip の管理

Zip エディタは通常のエディタと同じように文字列の編集も行うが、引用関係の管理と変更履歴の管理が特徴である (図 8)。 Zip で管理されたドキュメントがうさわ話と異なるのは、原著者と変更履歴のトラッキングがされている点である。図 8の「竜の落子」にある尻尾は、変更履歴を表す。尻尾 tail は tale と同じ発音で、 tale には歴史という意味がある。

Zip 上の操作は、以前の状態に戻るのに必要な情報とともに尻尾に記録される。このため、操作のやり直し／再実行が可能である。新しい Zip を作るとき別の Zip の項目を引き継ぐなら、新しい Zip は別の Zip の最新状態を引き継ぐので尻尾はない状態にない。こうすることで、履歴情報でディスク・スペースが圧迫されることを軽減している。履歴情報を長期に取っておく文化は現在あまり見られないが、創作過程を保持しておくことは、どこが先に発明したか、本当に所有権をもっているかを争うようなケースに備えるためにも重要になってくるだろう。

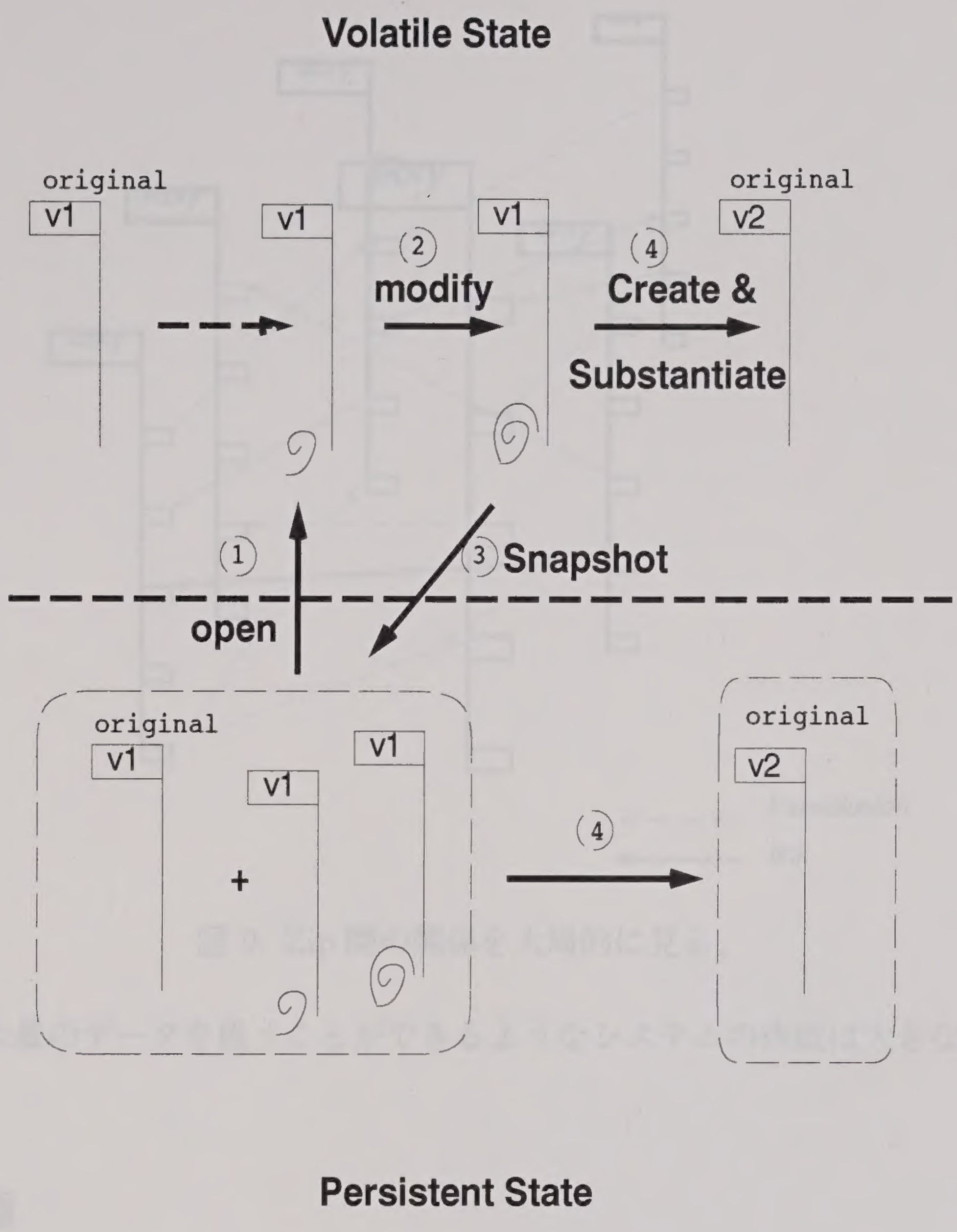

Volatile State
original
v1
v1
②
modify
v1
④
Create &
Substantiate
original
v2
①
open
③ Snapshot
original
v1
v1
+
v1
④
original
v2
Persistent State

図 8: バージョン管理

5　ユーザインタフェース

複数の Zip を並べて表示し、それらの関係を大局的に見る機能 (図 9) は大変有力なツールになる。例えば、裁判の記録を取っているとすると、各 Zip は、原告、被告、証人の申し立てを表す。トランスクルージョンを使っている個所は、客観的事実や証言の一致している個所を示し、リンクは他の人の証言などを参照している個所を示す。

Zip は項目の並びであり、その項目は Zip を含むわけではないので、階層としては 1 レベルしかない。しかし、画面上に関連性の強い Zip を並べて、右にある Zip は左の Zip のより詳しい項目を含むものであれば、横方向に連係をとって、ワープロに見られるアウトライン・プロセッサとして機能させることができる。また複数の Zip から論理演算により別の Zip を作ることも計画されている。この機能により、複数の Zip から自動的に共通部分を抜き出したり、差を調べることが可能となる。

各 Zip や項目はパッドに対応させようとしている。こうすることで IntelligentPad の他のパッドの機能を使ったり、他のパッドで Zip の機能が使えるようになると期待される。 Zip エディタのユーザ第 1 号は Ted 氏自身の予定である。これまで蓄積してきた膨大な記録を入力したいというのが彼の夢である。彼が語ったところによると項目数で 50 万件ほどのデータがあるそうであ

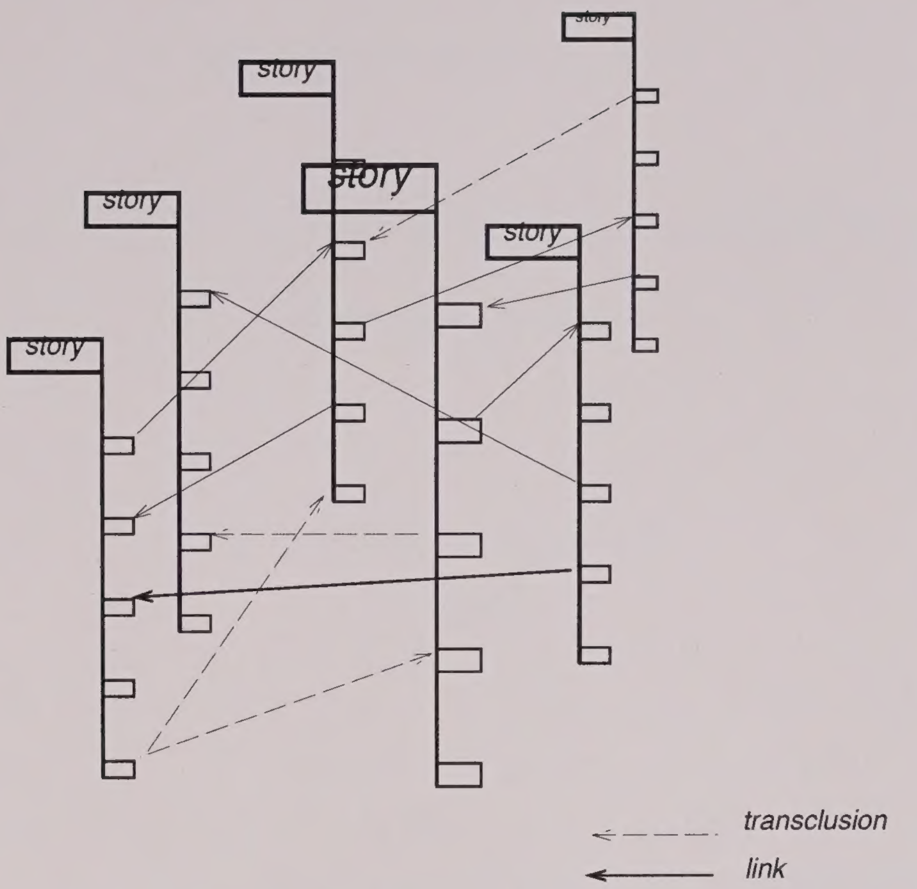


図 9: Zip 間の関係を大局的に見る。

る。このような大量のデータを扱うことができるようなシステムの作成は大きなチャレンジと言えるだろう。

6　将来の計画

現在はスタンドアロンの Zip エディタを開発中であるが、ネットワーク出版においてこそトランスクルージョンは真価をはっきする。将来はクライエント／サーバ方式で、実際にアクセスされたページ分原著者に支払われるような仕組みも検討したいと思っている。

また WWW がたいへんな盛り上がりを見せているので、これと併せて使えるようにすると大変役に立つツールとなる。そこで、HyperLab のプロジェクト 1 で作っている HTML ビューアーのページを Zip に貼ったり、逆に Zip の項目をビューアーに貼って HTML ドキュメントにできるようにすることも検討したいと思っている。さらに現在は項目としてテキストしか考えていないが、イメージやグラフィックスも扱えるようにしたいと考えている。

参考文献

[1] Theodor H. Nelson, *A File Structure for The Complex, The Changing and the Indeterminate*, Association for Computing Machinery, 1965

[2] Theodor H. Nelson, *The Heart of Connection: Hypermedia United by Transclusion*, Communication of the Association for Computing Machinery, to be published in 1995

[3] Theodor H. Nelson, *HyperLab Zip Editor: Preliminary Specifications*, March 1995

OPEN HOUSE March 22, 1995

ネットワーク上に展開するPad流通機構のDemonstration

日立製作所 中央研究所
日高 稔

はじめに

Sapporo Hyperlabでは、World Wide Web(WWW)を利用した情報提供機構の商用利用に対する検討を行っている。情報提供機構としてのWWWの有効性は、現状のInternetの普及の牽引力となっていることからも、明らかである。WWWがこれほど普及した要因は、ユーザーインターフェイスであるMosaic等のWWW viewerの操作性の簡便さと、Internet Protcolが他の場所にある情報へ簡単にアクセスできる機構を備えていたことにあるといっても良いだろう。特にHyperTextによる他の情報とLINKできる機構は新たなInternetの形態を与えたと言っても過言ではないだろう。

この簡便なユーザーインターフェイスとInternetの商用利用解放とが、計算機の機能に明るくない一般ユーザーを著しく増加させているが、今後、彼等がInternet利用の主たる担い手になることは容易に想像ができる。このレベルのユーザー増加こそがInternetの商用利用に拍車をかけることは間違いない。彼らは一般にプログラミングには明るくなく、またその必要もないと考えている。しかし、既存のプログラムのカスタマイズは多くのユーザが望んでいる。最近ではある程度のカスタマイズをサポートしたアプリケーションプログラムが発売され始めた。

その代表として、プログラミングは必要なく、カスタマイズは自由にでき、しかも、それを簡単にするためにPadとよぶ部品を組み合わせるだけで目的とするアプリケーションが組み立てられるとのコンセプトから開発されたIntelligent Pad(IP)がある。IPでは、Padを組み合わせて目的とするアプリケーションを組み立てるが、その部品の流通にInternetを用いることでWorld WideなPad流通機構の形成を考えている。

Sapporo HyperlabではProjectの一つとして、Inteligent Padのアプリケーションとして、WWW viewerを開発すると同時に、提供する情報としてのPadの流通を検討している。

開発目標

Pad流通機構のユーザーインターフェイスの開発

Intelligent Padを用いたWWW viewerを開発する。

これは、現状のMosaic,Netscape等のWWW viewerの上位互換となる必要がある。つまり、現状のWWWの普及状態を顧みると、今後開発する情報提供機構はWWWとの互換性を無視することはできないため、それに応じた機構の開発が不可欠である。

特徴としては取り込んだviewer上の情報をPadとして分解、統合できる機能を備えることで、現状のWWW viewerに対する優位性を持つものとする。

情報提供機構

ユーザー認証機構、課金機構を備えたWWW serverの検討。

商用利用を現実性のあるものにしようとした場合、情報提供機構のセキュリティと課金機構の検討は不可欠である。そこで、現状のWWW serverの持つセキュリティを強化したサーバーの構成、実用性のある課金機構の選択について検討する。

Demonstration 概要

情報は図１の様にInternetを介して接続されたWorkstation上に分散して存在している。

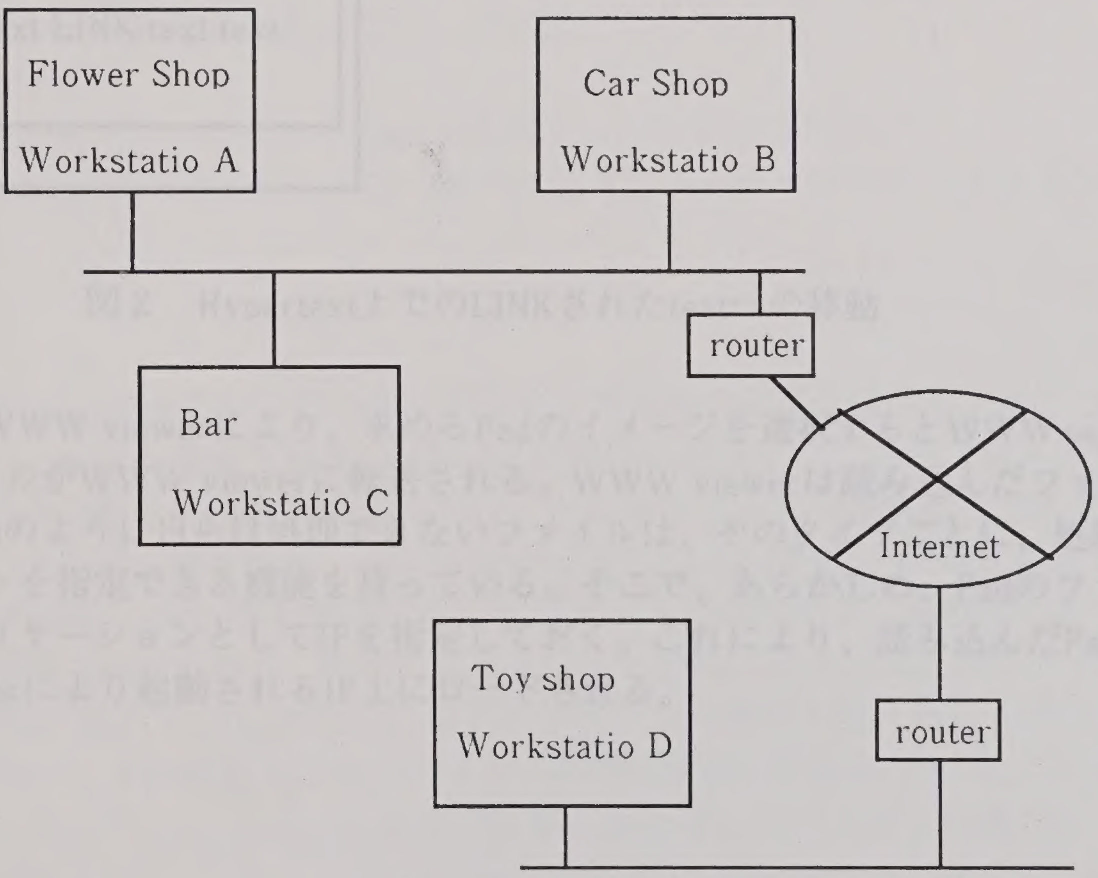

Flower Shop
Workstatio A
Car Shop
Workstatio B
router
Bar
Workstatio C
Internet
Toy shop
Workstatio D
router

図１　WWW serverの分散環境

その状況を複数のWWWserverを立ちあげることでHyperlab内に実現する。

将来的には、分散した複数のWWW server上に開かれたshopから、求めるPadを読み込むとともに清算機構を実装した情報提供機構を実現する。

今回はその操作状況を既存のWWW viewerを利用してDemonstrationする。

Mosaic等のWWW viewerを起動すると、初めにその環境のHomepageが表示される。そこからHypertext中のLINKをたどることで、World Wide Webの中を移動できる。

Mosaic等のWWW viewerでは、表示したHypertext中の色のついたテキスト、あるいは色つきの縁どりのあるイメージが、他のHypertextにLINKしていることを示す。その色つきのテキストあるいはイメージをMouse等のpointingdeviceでclickすると、LINK先のHypertextが表示される。

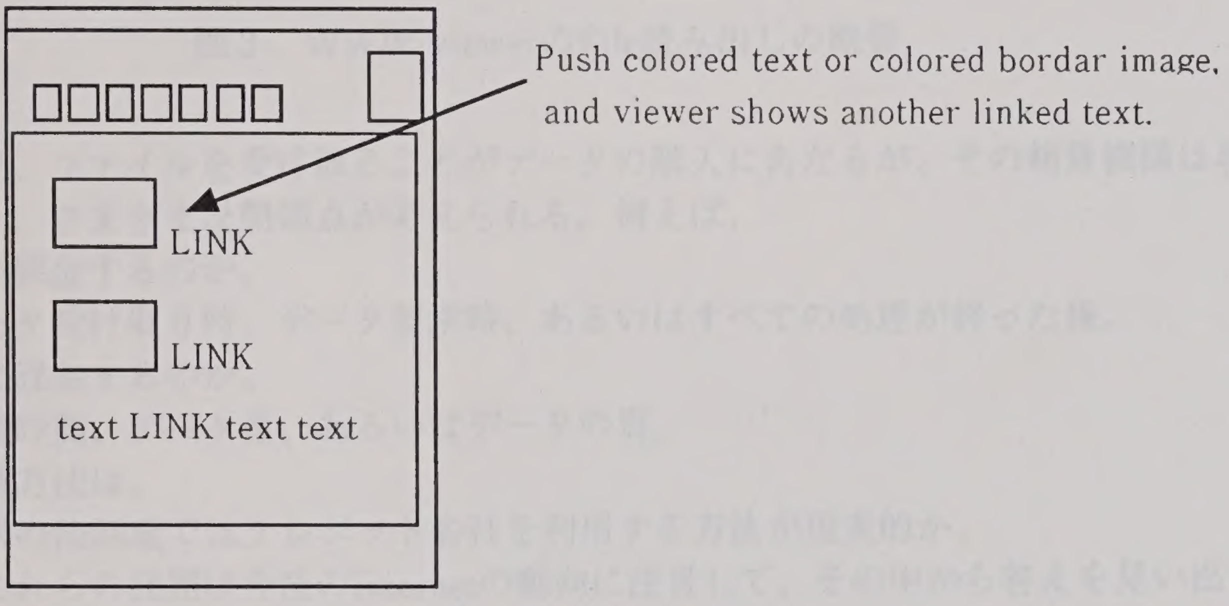

Push colored text or colored bordar image,
and viewer shows another linked text.
LINK
LINK
text LINK text text

図２　Hypertext上でのLINKされたtextへの移動

そこで、WWW viewerにより、求めるPadのイメージを選択するとWWW server上のPadのファイルがWWW viewerに転送される。WWW viewerは読み込んだファイルを表示するが、Padのように自らは処理できないファイルは、そのタイプごとに、処理するアプリケーションを指定できる機能を持っている。そこで、あらかじめ、Padのファイルに対応するアプリケーションとしてIPを指定しておく。これにより、読み込んだPadは、WWW viewerにより起動されるIP上にロードされる。

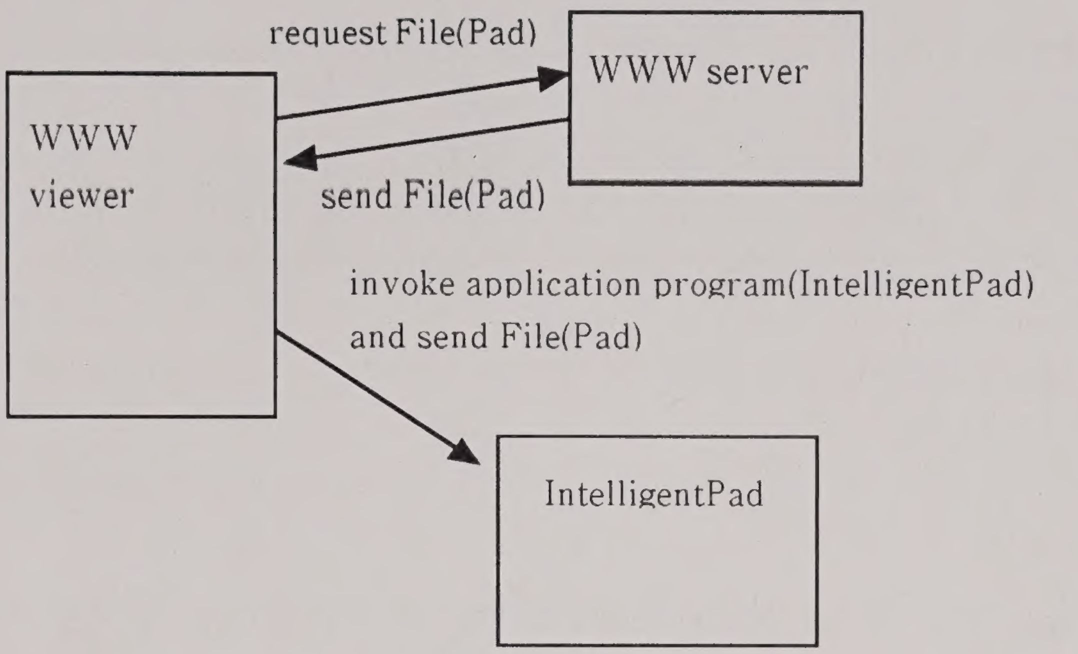

request File(Pad)
WWW server
WWW
viewer
send File(Pad)
invoke application program(IntelligentPad)
and send File(Pad)
IntelligentPad

図３　WWW viewerのFile読み出しの概要

ここで、ファイルを受け取ることがデータの購入にあたるが、その精算機構は現在検討中であり、さまざまな問題点が考えられる。例えば、

1 いつ課金するのか。
　データ受け取り時、データ要求時、あるいはすべての処理が終った後。
2 何に課金するのか。
　接続時間、データ量、あるいはデータの質。
3 決済方法は。
　現状の法制度ではクレジット会社を利用する方法が現実的か。

など。これらの課題は今後のInternetの動向に注目して、その中から答えを見い出す必要がある。

さいごに

今回のDemonstrationでは、これから開発する情報提供機構を既存のプログラムを使って概説した。これらの機能を実装したPad流通機構の実現には、さらに多くの課題があると思われる。特に企業のLANとの接続では、そのセキュリティが問題となるだろう。

既に述べたように、基本的に、WWW server(NCSA httpd等)は、Disk上のファイルをクライアント(WWW viewer等)の要求に応じて送出する機構である。現状でもある程度のセキュリティ機構は持っており、現状のWWW serverでは、そのDisk上のファイルは、その接続要求元のIntenet Protcol addressに応じて、アクセス権限を変えることができる。最近のInternet上では、その機構を用いたアクセス制御の実験が行われている。Hyperlabにおいても、ローカルドメインsec.or.jpからの接続しか許可しないといった簡単なアクセス制御を行っている。

今後は、この機構を用いた高度なアクセス制御を検討し、さらにセキュリティの高いシステムの構築を目指したい。